A Statistical and Machine Learning Approach to Air Pollution Forecasts

Simon Carlén

Degree project, 15 credits
Computer and Systems Sciences
Degree project at the master level
Spring term 2022
Supervisor: Sindri Magnússon
Co-supervisor: Ali Beikmohammadi
Reviewer: Petter Karlström

Stockholm University

Abstract

In today's world, ambient air pollution is a pressing environmental concern, as the majority of the world's population is regularly exposed to air pollution levels higher than what is deemed healthy. To mitigate harmful effects, city air is regularly monitored, and such monitoring can produce massive amounts of data. This enables the development of statistical and machine learning techniques for detailed air quality forecasts. The data from multi-sensor air monitoring systems can be challenging to utilize though, as there are dependencies over both space and time. However, for this type of problem, deep neural network models have shown promising results, and in this work, several neural network architectures, as well as a more straightforward multiple linear regression technique, are developed to forecast nitrogen dioxide levels at a forecast horizon of one hour. For several evaluation metrics, the neural network models outperformed the multiple linear regression model. However, none of the models had the desired structure of the forecast errors, and all models failed to successfully capture sudden pollution peaks. Nevertheless, the results point to an advantage for the more complex neural network techniques, and further advances in the field of deep learning, together with higher resolution data, have the potential to improve air quality forecasts even more and cross conventional limits.

Keywords: Air pollution forecasts, neural networks, linear regression, time series analysis, nitrogen dioxide

Synopsis

Background To mitigate the harmful effects of air pollution, air quality is regularly monitored. Such monitoring produces massive amounts of data, and this enables the development of statistical and machine learning techniques for modeling and forecasting of air pollution.

Problem Air pollution is a complex phenomenon depending on many factors, and the data from air monitoring has both temporal and spatial dependencies. This makes modeling and forecasting a challenge, and the research problem in this work is: *To capture and model the complex dynamics of air pollution with machine learning methods, with an emphasis on deep learning.*

Research Question The research question for the thesis is: *How can machine learning, in particular deep learning, be used to forecast air pollution levels and pollution peaks?* The research question emphasizes pollution peaks, as these are much harder to accurately predict and forecast than pollution levels at lower ranges.

Method Data was downloaded from SMHI's centralized database for air measurements, carefully examined, and thereafter preprocessed before a linear regression and several deep learning models were fit. All models were trained with historical data and later evaluated on the most recent (test) data. An important aspect of model evaluation was a close examination of the forecast errors. The Python programming language was used together with libraries for scientific programming and data science/machine learning.

Result All deep learning models outperformed the linear regression model. However, for all models, the structure of the forecast errors were not as desired, and this warrants further model refinements. The structure of the errors were due to the inability of the models to capture pollution peaks well. Though the deep learning models showed promising potential, in light of the research question, the results were unsatisfactory.

Discussion The forecast horizons in this work (one hour) are very short-term, which limits usability. To this end, however, adaptations to extend the forecast horizons are possible. Further adaptations to include more than just one air pollutant is also an option; this would result in comprehensive forecasts that could be of great value to public health authorities and policymakers, as it could permit early interventions to protect public health, and vulnerable groups in particular.

Acknowledgements

First of all, I would like to give a large amount of thanks to my supervisor Sindri Magnússon, as well as my co-supervisor Ali Beikmohammadi, for guiding me through this project. Your help has been indispensable. I would also like to thank the reviewer, Petter Karlström, for valuable feedback on the first thesis draft. Lastly, I would like to thank my fiancée Anna for all the support and love you continuously give. Without you, this thesis would not have been possible.

Contents

List of Figures **i**

List of Tables **ii**

List of Abbreviations **iii**

1 Introduction **1**
- 1.1 Background . . . 1
- 1.2 Research problem . . . 1
- 1.3 Research question . . . 2
- 1.4 Delimitations . . . 2

2 Extended Background **3**
- 2.1 Ambient air pollution . . . 3
  - 2.1.1 Principal air pollutants . . . 3
  - 2.1.2 Ambient air pollution in Stockholm . . . 3
- 2.2 Forecasting air pollution . . . 4
  - 2.2.1 Forecasting as a regression problem . . . 4
  - 2.2.2 Linear regression models . . . 4
  - 2.2.3 More complex regression models and neural networks . . . 6
  - 2.2.4 Evaluating regression models and forecasting performance . . . 8

3 Methodology **10**
- 3.1 Data retrieval and preprocessing . . . 10
  - 3.1.1 Data sources . . . 10
  - 3.1.2 Data preprocessing . . . 11
- 3.2 Model fitting and hyperparameter tuning . . . 13
  - 3.2.1 Multiple linear regression models . . . 13
  - 3.2.2 Deep neural network models . . . 14

4 Results and Discussion **16**
- 4.1 Model evaluation . . . 16
  - 4.1.1 Common regression metrics . . . 16
  - 4.1.2 Examining the forecast errors . . . 16
- 4.2 Inference for the regression metrics . . . 18
- 4.3 Summary and discussion . . . 19

5 Conclusions **21**

6 Bibliography **22**

Appendices **25**

A Monitoring stations . . . 25

B Model diagnostics and summary statistics for the multiple linear regression models . . . 26

C Histogram and qq-plots of the forecast errors, Box-Pierce test results, and results from the Dunn's test . . . 29

List of Figures

2.1 An artificial neuron illustrated. 7

3.1 NO_2 data for Torkel Knutssonsgatan. 11
3.2 Temporal variables for day as sine and cosine waves. 12
3.3 Rolling window approach for time-series data. 13
3.4 RMSE for each candidate model generated by RFE. 14

4.1 Performance metrics for all models. 17
4.2 Scatterplots of observed vs. predicted NO_2 values. 18
4.3 Observed and predicted NO_2 levels during 10 days in December. 19

B.1 Residual plots for the OLS regression model. 26
C.1 Normal probability plots for the forecast errors. 30
C.2 Histogram for the forecast errors. 30

List of Tables

3.1 The best set of hyperparameters for each model. (NA = not applicable.) 15

A.1 Monitoring stations. 25
B.1 OLS regression results. 27
B.2 Robust linear model regression results. 28
C.1 Results from the Box-Pierce tests. 29
C.2 *p*-values for the pairwise comparisons from the Bonferroni-Dunn post-hoc test for the MSE. 29

List of Abbreviations

ANOVA	Analysis of variance
ACF	Autocorrelation function
EHA	Environment and Health Administration
fcNN	Fully connected neural network
GRU	Gated recurrent unit
LSTM	Long short-term memory
MAE	Mean absolute error
MAPE	Mean absolute percent forecast error
ME	Mean error
MSE	Mean squared error
MLR	Multiple linear regression
OLS	Ordinary least squares
RNN	Recurrent neural network
RFE	Recursive feature elimination
RMSE	Root mean squared error
PM	Particulate matter
QQ	Quantile-Quantile
SGD	Stochastic gradient descent
SMHI	Swedish Meteorological and Hydrological Institute
SVR	Support vector regression
SLB-analys	Stockholms Luft- och Bulleranalys
VOCs	Volatile organic compounds
WHO	World Health Organization

1. Introduction

1.1 Background

Outdoor air pollution is a major global environmental issue, linked to several serious health conditions, and causing millions of premature deaths every year [1]. Some principal air pollutants damaging to health include gaseous substances such as nitrogen oxides (NO_x), ground-level ozone (O_3), sulphur dioxide (SO_2), and carbon monoxide (CO), but also atmospheric aerosol particles such as PM_{10} and $PM_{2.5}$ [2]. In Stockholm, traffic is a major source of local air pollution, and though air quality is generally good, some streets experience short episodes with severe pollution levels, especially during winter and spring [3].

To protect public health, urban air is normally monitored. In addition to monitoring, forecasts of air quality (both hourly and daily) can be critical to regulatory authorities. In general, there are two approaches to such forecasts; mechanistic models and statistical/machine learning models. [4–6]. With mechanistic models, the processes governing the evolution of air pollution is modeled mathematically, whereas statistical and machine learning models are much more data-driven [6].

From a statistical perspective, predicting air pollution levels is a time series regression problem, and there are many different regression techniques for modeling and forecasting time series [6]. These techniques can vary in complexity, from more simple linear models to deep neural networks capable of finding complex non-linear relationships in the data [6, 7]. Nonetheless, one of the main challenges with air pollution is that there are dependencies over both space and time, and simpler models may not capture these dependencies well [5]. Recent advances in machine learning however has shown promising results with air quality forecasts, especially deep neural networks "tailored" to sequence and time series data [5, 6]. In this work, several deep learning architectures as well as more straight-forward linear model are explored for making hourly predictions of a commonly measured air pollutant, namely NO_2.

1.2 Research problem

Forecasts, be it for weather, stock returns, or future pandemics, are always associated with uncertainty and errors. Erroneous predictions made by existing air pollution forecasting systems, both mechanistic and statistical and/or machine learning-based, can be attributed to many causes. In the case of mechanistic models, there can be insufficient information in terms of the factors needed for simulation and modeling [6]. For statistical and/or machine learning methods, too simplistic models, lack of data, irrelevant input features, overfitting, etc., can limit prediction accuracy [6]. Nevertheless, atmospheric pollution is a very complex phenomenon depending on a multitude of factors across both space and time. Hence, the research problem addressed in this work is: *To capture and model the complex dynamics of air pollution with machine learning methods, with an emphasis on deep learning.*

1.3 Research question

From a forecasting perspective, of special interest are episodes when pollution levels peak. Generally, this is also when existing forecasting systems tend to give the largest prediction errors [6]. Therefore, the research question this thesis tries to answer is: *How can machine learning, in particular deep learning, be used to forecast air pollution levels and pollution peaks?*

1.4 Delimitations

In this work, historical air pollution and weather data is used. Therefore, the models are not tested in operational mode, i.e., with real-time data to make predictions, and consequently only the theoretical performance of the models can be known. The data for which the models were evaluated on are also from a time period still affected by the COVID-19 pandemic, with relatively low air pollution levels. Since forecast errors are generally larger when pollution levels are higher, there is a risk of getting overly optimistic prediction results, and this should be kept in mind when contemplating the performance of the models.

2. Extended Background

2.1 Ambient air pollution

Ambient air pollution is one of the greatest environmental and health concerns of the modern world. Worldwide, poor air quality causes millions of premature deaths every year and is linked to several adverse health effects such as respiratory problems, cardiovascular disease, and cancer [1]. In addition to health risks, the global economic impacts are substantial due to lost labor productivity, increased health care costs, reduced crop yields, etc. [8]. Outdoor air pollution has become a ubiquitous problem, affecting both cities and rural areas, and it is estimated that about 90% of the world's population are living in regions where air pollution levels exceed guidelines set by the World Health Organization (WHO) [1].

2.1.1 Principal air pollutants

In densely populated urban areas, air pollution levels can periodically be severe, and with an accelerating urbanization, it has become imperative for public health authorities to closely monitor city air and try to mitigate the harmful effects of pollution. Commonly monitored substances include sulphur dioxide (SO_2), nitrogen oxides (NO_x, i.e., NO and NO_2), carbon monoxide (CO), ground-level ozone (O_3), volatile organic compounds (VOCs), and particulate matter (PM) [2]. Vehicular traffic is a major source of the gaseous pollutants NO_x, SO_2, CO, and VOCs, but certain industrial processes also contribute to emissions [2]. Ground-level O_3 (also a gas) is a so-called secondary pollutant that forms when NO_x and VOCs react on sunny days with little wind [2]. PM are atmospheric aerosol particles (particles suspended in the air). They have diverse origins, both natural and anthropogenic, and a complex chemical composition consisting of both solid and liquid species [2].

2.1.2 Ambient air pollution in Stockholm

In the city of Stockholm, environmental air quality standards are usually met, though some streets experience occasional episodes with severe pollution levels (e.g. Hornsgatan is one such street) [9]. Since Stockholm has centralized district heating and few industries, the major source of local CO, NO_x, and PM pollution is vehicular traffic [3, 9]. Mechanical wear by studded tires on asphalt and the wearing of brakes and tiers in motor vehicles contribute substantially to local levels of both PM_{10} and $PM_{2.5}$. For $PM_{2.5}$ however, contribution from sources outside of Stockholm is also significant [9]. Emission of SO_2 can come from the energy sector and waterborne transport, though local levels are also affected by outside sources. For O_3, long-range transport from mainland Europe is the single-most important factor contributing to locally measured levels [9].

The air in Stockholm County is monitored by Stockholms Luft- och Bulleranalys (SLB-analys), a unit in the Environment and Health Administration (EHA) of the city of Stockholm. SLB-analys are responsible for a number of monitoring stations measuring several air pollutants and some meteorological parameters in the Stockholm region, as well as a few stations outside of Stockholm [10]. In addition to monitoring the air, SLB-analys also model and forecast air pollution levels for the Stockholm metropolitan area, and their forecasts are available through a smartphone application, called "Luft i Stockholm" [3].

2.2 Forecasting air pollution

Having the possibility to forecast air pollution levels hours or days ahead can be extremely valuable to regulatory authorities in order to protect public health, and vulnerable groups in particular. In general, there are two broad categories of models for such forecasts; mechanistic models, and statistical and/or machine learning models [4]. This work is concerned with the latter type, and in the sections below a review follows. The mathematical and statistical theory behind many of the models is quite extensive [7, 11–13], but relevant theory will be covered briefly.

2.2.1 Forecasting as a regression problem

While mechanistic models are based on mathematical modelling of atmospheric processes along with other factors governing the distribution of air pollution (such as emission source characteristics, physico-chemical properties of pollutants, terrain and building design, etc.) statistical and/or machine learning models are entirely data-driven, being derived directly from measurements on the variables of interest [4].

From a statistical learning perspective, forecasting air pollution can be viewed as a regression problem, in which a function f, mapping input data to a numerical output, is being approximated (or learned) from a training set of labeled input-output examples [13]. Learning the function f amounts to finding a set of parameters (or weights/coefficients) for the model, which in the case of a simpler regression technique can be only a handful, but possibly millions if a deep neural network is used [7]. Generally in regression, the weights are learned by minimizing a cost function

$$J(\hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \left(\hat{y}_i - y_i\right)^2 \tag{2.2.1}$$

where $\hat{\boldsymbol{\beta}}$ is a vector of estimated model parameters ($\hat{\beta}_0$, $\hat{\beta}_1, ...,$ $\hat{\beta}_n$), $\hat{y}_i$ is a prediction, and y_i is an observed value [13]. Depending on the regression model, minimizing $J(\hat{\boldsymbol{\beta}})$ with respect to the model parameters is approached differently, as explained further in the sections below.

2.2.2 Linear regression models

From the wealth of available regression techniques, multiple linear regression (MLR) has been extensively used to forecast and model air pollution [6]. Generally, if none of the basic model assumptions are violated, MLR is a straightforward method. However, air pollution monitoring typically produces time series data, for which the assumption of independent errors is often not appropriate [14].

Linear regression for time series data

If fitting a MLR model to time series data, successive errors will typically be correlated (often referred to as autocorrelation), and this will cause several problems with the model if the correlation is not accounted for [12]. To this end, adjustments to the MLR model can be made, some of which will require other parameter estimation techniques than the standard method of ordinary least squares (OLS). However, a simple and commonly used procedure to eliminate the autocorrelation is to include one or more lagged values of the response variable as predictors. For example, if the value of the response variable at lag one (y_{t-1}) is included, the MLR model will have the form

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 x_{2,t} + ... + \beta_k x_{k,t} + \varepsilon_t, \quad t = 1, 2, ..., T \tag{2.2.2}$$

where ε_t is the the error term, and t denotes time steps [12]. The model in Eq. (2.2.2) can be fit with OLS, which in linear regression is the standard way of finding model parameters so that $J(\hat{\boldsymbol{\beta}})$ is minimized [14]. This is done by solving the so-called least squares normal equations, and the least squares estimates of the model parameters are then given by Eq. (2.2.3) below, where X is a matrix of regressor variables and y is a vector of response variables.

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{2.2.3}$$

A commonly used test for detecting autocorrelation is the Durbin-Watson test, where the statistic will have a value of ~ 2 in the case if uncorrelated errors [12].

Robust linear regression models

The errors of a MLR model should ideally be independent, have constant variance, and be approximately normally distributed [14]. For inference and prediction, the normality assumption is important. Deviations from normality can sometimes be reasonably ignored, however, when the error distribution has long (or heavy) tails, this can be a sign of many extreme values in the data, in which case so-called robust estimation techniques are more appropriate than OLS [14]. Typically for air pollution data, there are recurring periods where extreme values are frequent (for NO_2 this often happens during winter months, see Fig. 3.1 on page 11), and for this reason, robust regression might be better suited than OLS regression for modeling and forecasting.

One robust estimation technique is M-estimation, where a modified cost function is minimized to find the best parameter estimates:

$$J(\hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \rho\big(\hat{y}_i - y_i\big)^2. \tag{2.2.4}$$

In Eq. (2.2.4), ρ is a so-called robust criterion function, for which there are several alternatives, but a popular choice is the Huber's t function (or Huber's method), where

$$\rho(z) = \begin{cases} \frac{1}{2}z^2, & \text{if } |z| \leq t \\ |z|t - \frac{1}{2}t^2, & \text{if } |z| > t \end{cases}$$

and where t is a robust estimate of σ [15]. In general, minimizing Eq. (2.2.4) is done iteratively by solving weighted least squares normal equations and (at each iteration) recomputing the weights until convergence is reached [14]. In effect, this makes the parameter estimation procedure much less sensitive to outliers and extreme observations in the data.

Additional considerations for linear models

Careful variable selection in MLR is also crucial as it can influence the performance of a model, and one is often concerned with finding an optimal "subset" of predictors, where multicollinearity should also not be an issue [14]. To this end, variable selection techniques based on optimizing a quantity of interest, e.g. the Akaike information criterion or the root mean squared error (RMSE) are common, and a diagnostic for multicollinearity is the condition number, where a value below 100 does not indicate any serious problems [14].

The extensive use of MLR for air pollution forecasts is many times motivated by its simplicity and straightforward implementation [6]. Another advantage is interpretability; for example, inference can be made on all input variables, allowing one to investigate their individual importance and relationship to the response variable [14]. However, the statistical properties of MLR make it rather restrictive as a model, and not all violations of the assumptions can be remedied (like non-linearity) [12]. More flexible regression models, better suited to capture complex input-output relationships, have also found extensive use in air pollution forecasts [6], and in the next section, some of these models are reviewed.

2.2.3 More complex regression models and neural networks

For air pollution, more flexible regression techniques tend to give better forecasting results than the more simplistic linear models [6]. Some examples include regression trees, support vector regression (SVR), and artificial neural networks [6, 16]. In many ways, these regression techniques are extensions to the linear model, e.g., SVR is built on the idea of constructing non-linear transformations of the regressors, whereas the building blocks of artificial neural networks are what is often called (artificial) neurons, where the output of a linear regression model is transformed through a non-linear activation function [13]. In this work, a number of neural network architectures are utilized, and a deeper dive into the theory behind them is warranted.

The artificial neural network model

As described above, in an artificial neuron, the output of a linear regression model is passed through a non-linear activation function; some choices for activation functions include the rectified linear unit (ReLU), or the tanh function:

$$\mathrm{ReLU}(x) = \max(0, x) \tag{2.2.5}$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.2.6}$$

A common way to illustrate an artificial neuron is shown in Fig. 2.1, where the inputs (including the so-called bias term, which is always equal to 1), are multiplied by the corresponding weights (w), and where the linear combination (z) subsequently gets transformed by the activation function f in the node before an output is produced.

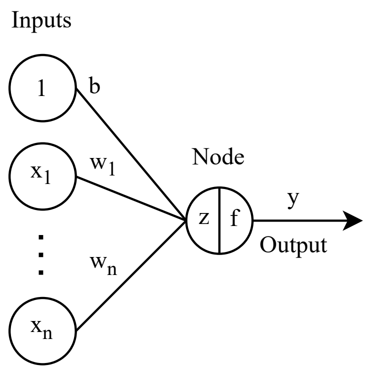

Figure 2.1: An artificial neuron illustrated.

A layer in a neural network is made up of several nodes (also called hidden units), and in a deep neural network, many such layers can be stacked sequentially, so that the outputs of the nodes in one layer becomes the input to the nodes in the next [13]. Artificial neural networks can have many thousands or even millions of trainable parameters/weights, which makes them extremely flexible. However, this comes at a great computational cost, and neural networks are also very prone to overfitting, which is why regularization techniques frequently have to be used (described further below) [13].

Deep learning architectures

The most basic type of neural network model is the fully connected neural network (fcNN). (This is also sometimes called a densely connected neural network, or simply, a dense model). A fcNN basically has the structure of the neural network model described above, i.e., with nodes in layers stacked sequentially, and where at each layer, a slightly more abstract representation of the inputs are produced by non-linear transformations. With several such transformations, complex relationships in the data can often be learned [7, 13].

For data where order matters (like with sequence and time series data), so-called recurrent neural network (RNN) models are often more appropriate than the fcNN [7]. RNN's can utilize past information in a sequence through a so-called 'state vector', which works like a memory, holding the information from previous time steps. The state vector is also continuously updated as a data sequence is being processed, so the output at any time t depends on the current input as well as the information contained in the "memory" (or the systems hidden state) [7]. The most basic RNN architecture, the simple RNN model, can be difficult to train, however, and typically also performs poorly when processing longer sequences due to the way the state vector is maintained and updated [7]. For this reason, a much more common RNN model is the so-called long short-term memory (LSTM) model. The LSTM model utilizes three so-called gates (the input gate, the output gate, and the forget gate), which together with an additional vector (the cell-state vector), controls the information flow through the units of the network [7]. Yet another common RNN model is the gated recurrent unit (GRU) model, which is similar to the LSTM model but with fewer parameters [17].

Training artificial neural networks

The principal method for finding the best parameters when the cost function is non-convex (as with neural networks) is the gradient descent algorithm [13]. This algorithm iteratively updates the parameters by, at each iteration, adjusting the parameter vector by a small amount in the opposite direction to the gradient of the cost function [13]. How much the parameters are adjusted at each iteration is decided by a hyperparameter referred to as the learning rate (usually a small number). For large datasets, computing the gradient for the cost function can be problematic, which is why the stochastic gradient descent (SGD) algorithm can be used instead [13]. SGD is very similar to gradient descent, except that subsamples of data are used at each iteration instead of the entire dataset. Also with SGD, due to the gradient being estimated (with only part of the data), the learning rate cannot be constant but instead has to be adapted to gradually decrease [13]. As mentioned above, the gradient descent algorithm utilizes the gradient of the cost function, which is calculated with the so-called backpropagation method [13].

The many trainable parameters of neural networks make them prone to overfitting, and a common technique to counteract this tendency is dropout [13]. Dropout randomly selects units in the network and removes them, thereby also removing all connections to those units. The main idea behind dropout is to make the network fit a little less well to the training data, but as a result of that, generalize better to new data [13].

2.2.4 Evaluating regression models and forecasting performance

Common regression metrics

When evaluating the performance of a forecasting model, one is often concerned with the one-step ahead forecast error, defined as

$$e_t(1) = y_t - \hat{y}_t(t-1)$$

where $\hat{y}_t(t-1)$ is the forecasted value of y_t made at time $t-1$, and accuracy metrics such as the mean error (ME), the mean absolute error (MAE), the mean squared error (MSE), and the mean absolute percent forecast error (MAPE) are often utilized [12]. These performance metrics are standard when evaluating regression models, and the equations for them are not repeated here. It can be noted though that the ME is an estimate of the expected value of $e_t(1)$, and strong departures from zero would indicate bias in the forecasts [12]. Also, commonly the root mean squared error (RMSE) is used in connection with the MSE, as the MSE gives the magnitude of the errors in squared, and not original, units. The MSE is an estimate of the variance of $e_t(1)$, and consequently the RMSE becomes an estimate of the standard deviation [14].

Hypothesis testing and inference

Hypothesis tests can also be carried out for the regression metrics, where one is concerned with whether the differences between competing models hold, or are simply due to chance. If the same dataset is used for many models, a suitable test is the one-way repeated measures analysis of variance (repeated measures ANOVA) test [18], where the differences in e.g. the MSE between the models can be compared. The repeated measures ANOVA test requires normality though, and a non-parametric alternative is the Friedman's test [18]. Following the repeated measures ANOVA or the Friedman's test, pairwise comparisons can be carried out with e.g. Tukey's test or the Bonferroni-Dunn test which are, respectively, a parametric and a non-parametric test [19].

Tests for autocorrelation in forecast errors

An important concept in time series analysis is that of autocovariance and autocorrelation [12]. The autocovariance and the autocorrelation coefficient for two observations k lags apart in a time series are given, respectively, by Eq. (2.2.7) and Eq. (2.2.8) below.

$$\gamma_k = \mathrm{Cov}(y_t, y_{t+k}) = E[(y_t - \mu)(y_{t+k} - \mu)] \tag{2.2.7}$$

$$\rho_k = \frac{E[(y_t - \mu)(y_{t+k} - \mu)]}{\sqrt{E[(y_t - \mu)^2]E[(y_{t+k} - \mu)^2]}} = \frac{\mathrm{Cov}(y_t, y_{t+k})}{\mathrm{Var}(y_t)} = \frac{\gamma_k}{\gamma_0} \tag{2.2.8}$$

The autocovariance function is the sequence of values of γ_k for $k = 0, 1, 2..., T$, and similarly, the autocorrelation function (ACF) is the sequence of values of ρ_k for $k = 0, 1, 2..., T$. In practice, these functions can only be estimated, and the estimates for the autocovariance function and the ACF (called the sample ACF) are given by Eq. (2.2.9) and Eq. (2.2.10), respectively.

$$c_k = \hat{\gamma}_k = \frac{1}{T}\sum_{t=1}^{T-k}(y_t - \bar{y})(y_{t+k} - \bar{y}), \quad k = 0, 1, 2, ..., K \tag{2.2.9}$$

$$r_k = \hat{\rho}_k = \frac{c_k}{c_0}, \quad k = 0, 1, 2, ..., K \tag{2.2.10}$$

In general, a time series is said to be white noise if the observations are uncorrelated and have constant variance. If the observations are also normally distributed, the time series is said to be Gaussian white noise, and ideally, this should be the case for the forecast errors [12]. If the forecast errors are white noise, the sample ACF will approximately follow a normal distribution with zero mean and variance $1/T$. To test whether any of a set of autocorrelations in the sample ACF are not zero (which then would indicate that the forecast errors are not white noise) the Box-Pierce test can be used [12]. Essentially, this is a goodness-of-fit test where it is determined how well the sample ACF fits to the ACF of white noise. A similar and commonly used test for smaller sample sizes is the Ljung-Box test [12].

Additional evaluation metrics

There are many additional ways in which the performance of a forecasting model can be evaluated. For example, in a study from 2018 by Pucer et al. [16], predictions are classified and cost matrices are utilized as part of model evaluation, and in a paper from 2007 by Stablober et al. [20], a quality function is defined in which each observation–forcast pair are rated, and where a larger penalty is also put on faulty forecasts in a specific range. In this work however, the common regression metrics are used together with a close examination of the forecast errors (using the methods described above) in order to answer the main research question of the thesis.

3. Methodology

The major steps of the implemented workflow were as follows; data were retrieved, examined, and preprocessed. Thereafter, a MLR model and several deep neural network models were fit to the data with the appropriate input variables. The models were then used to make forecasts for NO_2 with a one-hour forecast horizon. The NO_2 forecasts were limited to one station measuring background levels of air pollution in the center of Stockholm (Torkel Knutssonsgatan, see Table A.1 in Appendix A). Detailed descriptions of each step in the process are given in subsequent sections.

The programming language used was Python, together with several libraries for scientific programming, statistics, machine learning, and plotting/visualization (NumPy, pandas, statsmodels, scikit-learn, SciPy, TensorFlow and KerasTuner). All code can be found on the GitHub page[1] for the project (invite needed).

3.1 Data retrieval and preprocessing

3.1.1 Data sources

Air pollution data was retrieved from the Swedish Meteorological and Hydrological Institute's (SMHI) centralized database for air quality measurements [21]. This data is part of the national and regional environmental monitoring of Sweden, a program coordinated and funded by the Swedish Environmental Protection Agency (Swedish EPA) and the Swedish Agency for Marine and Water Management. There are in total ten different program areas, of which air is one, and all data are licensed under CC0 and therefore freely accessible to the public [22]. For the national air monitoring (under Swedish EPA's responsibility), SMHI acts as a national data host and stores (quality checked) historical data reported yearly from municipalities in Sweden [21].

Monitoring stations

In Stockholm County, there are 19 stationary sites for air pollution monitoring [10], and initially, data from each was considered. However, not all sites measure hourly NO_2, and for some sites the data were irregular. Therefore, data from four sites with hourly NO_2 measurements (in $\mu g/m^3$) for the time period 2016-01-01 to 2022-01-01 was subsequently chosen, giving a total of 52,609 data points. (However, as explained further below, a year worth of data, i.e., 8760 data points were subsequently excluded.) For the station at which NO_2 predictions were to be made (Torkel Knutssonsgatan), hourly meteorological data was also utilized. More specifically, these meteorological variables included temperature (in °C), precipitation (mm), atmospheric pressure (hPa), relative

[1] `https://github.com/simoncarlen/forecasting_project`

humidity (as %), solar radiation (W/m^2), and wind speed (m/s). The meteorological data were downloaded from SLB-analys' webpage [23].

In general, air pollution monitoring can be classified by the surrounding area (rural, rural-regional, rural-remote, suburban, and urban), and by the predominant emission sources (background, industrial, or traffic) [21]. The chosen stations included data from both traffic and background monitoring, in urban as well as rural-regional areas. More information about the stations are given in Table A.1 in appendix A.

3.1.2 Data preprocessing

Initial preprocessig

All stations had short episodes with missing data, and linear interpolation was used to fill in the missing values. Missing weather data was also linearly interpolated, except atmospheric pressure and wind speed for which mean imputation was deemed more appropriate. Moreover, before use in any of the models, all data were min-max normalized (i.e., scaled to the interval $[0, 1]$).

In Fig. 3.1, the NO_2 data for Torkel Knutssonsgatan is shown. A notable reduction in NO_2 levels can be seen during 2020 and early 2021; this reduction is most likely due to the COVID-19 pandemic, and by late 2021, pre-pandemic NO_2 levels are again approached. Because of this, a train-test split (see below) was done to entirely avoid using the data from 2020.

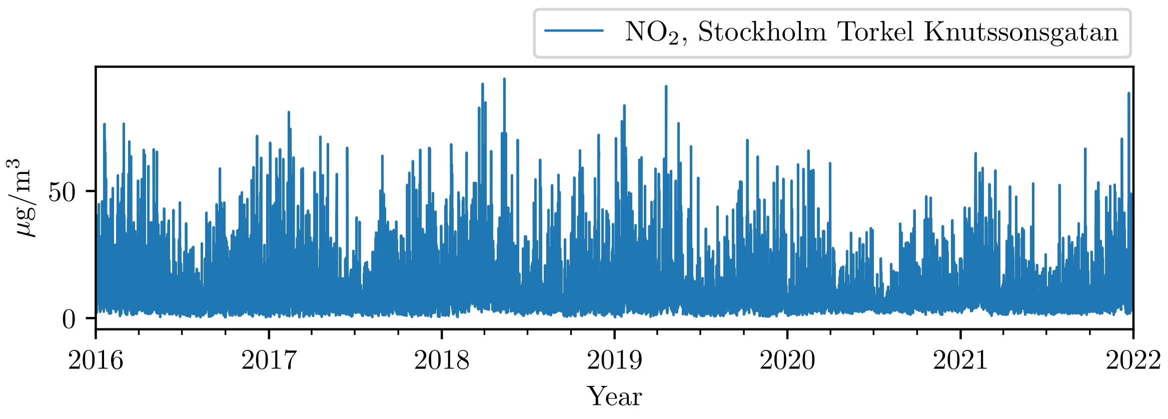

Figure 3.1: NO_2 data for Torkel Knutssonsgatan.

Creating temporal variables

In Fig. 3.1, yearly periodicity in the data can be seen, where levels tend to peak during winter months. Daily and weekly periodicity is also expected since traffic intensities vary throughout the day and week. To account for this, timestamps were converted to temporal variables as sine and cosine waves for day, week, and year. For example, the sine and cosine waves for day were calculated in the following way

$$\text{Sine day} = \frac{1}{2}\Big(\sin\Big(\text{timestamp} \cdot \frac{2\pi}{86,400}\Big) + 1\Big)$$

$$\text{Cosine day} = \frac{1}{2}\Big(\cos\Big(\text{timestamp} \cdot \frac{2\pi}{86,400}\Big) + 1\Big)$$

where timestamp is in UNIX epoch time[2] (and with 86,400 seconds in 24 hours, dividing by this term is necessary). The calculations were done similarly for week and year, except for the term in the denominator which instead was set to seconds per week and seconds per year, respectively. Note that the sine and cosine waves were adjusted to oscillate between zero and one. The temporal variables for day in a 24 hour time window are shown in Fig. 3.2.

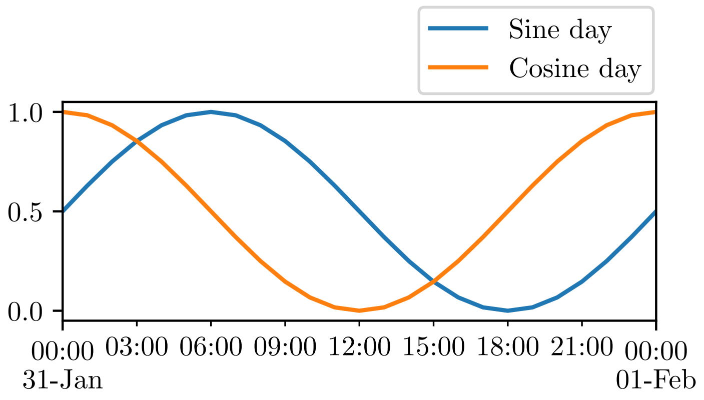

Figure 3.2: Temporal variables for day as sine and cosine waves.

Rolling windows

The rolling windows method extracts data sequences of certain lengths (the ”windows”) from the input data, and in each window is an ”input window” and a ”target window” [24]. For example, as shown in Figure 3.3 (where t indicate time steps), with a sequence of nine data points, the first eight observations would constitute the input window, and the ninth observation the target window. After extracting this sequence, a slide forward is made to extract the next sequence, and this is continued until observation n becomes the target window, at which point all the data have been processed.

In this work, rolling windows were used as input to the deep learning models, and different input window lengths were tested for making predictions of a target window one time step ahead (i.e., the next hour). More details are given in section 3.2.2 below.

Train-test split

Lastly, the data was split into training, validation, and test sets, where the validation set was used for hyperparameter tuning. The test set was taken as the most recent year of data (from 2021-01-25 to 2022-01-01, where the first 24 days of January were skipped due to many missing values at the Lilla Essingen station). For the validation set, the data from 2019 was used (since 2020 was a year with comparatively low air pollution levels). The remaining data was used for training (2016-01-01 to 2019-01-01). This ordered (as opposed to random) split is motivated by the time dependence in the data. It should also be noted that when normalizing the validation and test sets, min and max values from the training set were used. This ensures that model evaluation

[2]The UNIX epoch time for a given timestamp t is the number of seconds that has passed between January 1, 1970, and t.

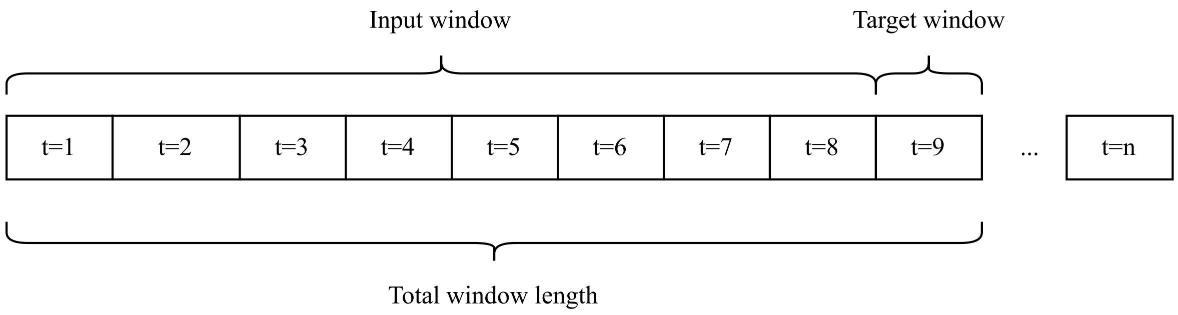

Figure 3.3: Rolling window approach for time-series data.

will be a good (and not too optimistic) measure of how well the models generalize to new, previously unseen, data points.

3.2 Model fitting and hyperparameter tuning

3.2.1 Multiple linear regression models

Initially, a simple linear regression model was fit with OLS where the response variable at lag one was used as predictor. No significant autocorrelation was seen with this model, but when also including the response variable at lag two as predictor, the Durbin-Watson statistic improved (i.e., was brought closer to 2). Including additional response variables after the first two lags did not lead to further improvements in terms of eliminating autocorrelation.

NO_2 data from other stations, meteorological variables, and the temporal variables were subsequently added to the model. These extra predictors did not lead to any serious multicollinearity, as indicated by the condition number. The NO_2 data was fit with the values at lag one, since for a forecast at time $t + 1$, these predictors cannot be known. However, lagged values of the meteorological variables were not used as these can more easily be replaced with their forecasted values. A similar MLR approach to the one taken here, but for predicting daily means of PM_{10}, can be found Stadlober et al. [20].

A log transformation of all NO_2 data was required before normalization to stabilize the variance of the errors, and also make the error distribution more normal. Even so, deviation from normality was indicated, and as can be seen in Fig. B.1 in Appendix B where OLS model diagnostics are shown, the error distribution had long/heavy tails. For this reason, a robust regression model with M-estimation (and Huber's t function) was judged to be a more suitable alternative to OLS regression.

At this point, with many input variables in the model, recursive feature elimination (RFE) was used as a variable selection technique, in which the model is repeatedly re-fit after having removed the least significant predictor [15]. Each candidate model generated by RFE was evaluated on the validation set (as well as the training set for comparison), and the results are shown in Fig. 3.4 on the next page.

From Fig. 3.4, it is evident that the least important predictors brought essentially no improvements to the model, and they were therefore removed. More specifically, these variables were; precipitation, atmospheric pressure, the Norr Malma NO_2 data,

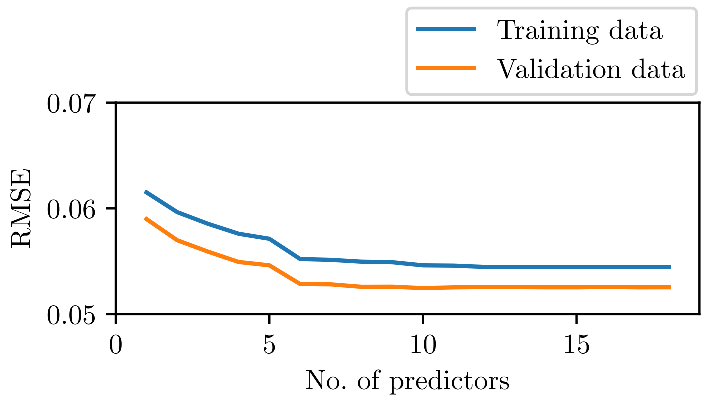

Figure 3.4: RMSE for each candidate model generated by RFE.

and the temporal variables for week and year. The final MLR model thus had the form

$$\log y_t = \beta_0 + \sum_{i=1}^{2} \beta_i \log y_{i,t-i} + \sum_{i=3}^{5} \beta_i \log x_{i,t-1} + \sum_{i=6}^{9} \beta_i z_{i,t} + \sum_{i=10}^{11} \beta_i w_{i,t} + \varepsilon_t \qquad (3.2.1)$$

where x denotes input variables with NO_2 data from other than the target station, z meteorological variables, w the temporal variables for day, and t = 1, 2, ..., T. Also, the errors (ε_t) in Eq. (3.2.1) are assumed to follow a Gaussian-shaped, heavy-tailed probability distribution (see Fig. B.1b in Appendix B). Summary statistics for this robust regression model, together with values and inference for the estimated parameters, are given in Table B.2 in Appendix B. Also, in Table B.1 in Appendix B, summary statistics for the OLS regression model are given, though this model was not used to make any forecasts.

3.2.2 Deep neural network models

Model fitting

At first, a fully connected neural network (dense model) was trained with the same input data as for the MLR model (i.e., the values at lag one and two for the response variable, lag one values for the NO_2 data from the other three stations, as well as the meteorological and temporal variables). Subsequent deep learning models, namely an additional dense model, a simple RNN, an LSTM, and a GRU model, were all fit with data windows of different input lengths (6, 12, and 24 hours, with the next hour as the target window), however with the same set of input variables as for the MLR and the first dense model.

It should be emphasized that these two model fitting strategies differ in that the models with rolling windows utilize multivariate time series from the immediate past, whereas the first dense and the MLR model utilize past pollution data, as well as "real-time" data of the meteorological variables (and hence the meteorological variables would need to be replaced with forecasted values should the models be used in operational mode). Having the same set of input variables (again, being NO_2 data from the target site as well as three adjacent sites, four meteorological variables, and the two temporal variables) for all models allows for a more direct comparison between the baseline MLR model and the deep learning models, as well as the two different model fitting strategies.

Hyperparameter tuning

For all deep learning models, the hyperparameters tuned were; number of hidden layers (up to five were tested), number of units in each hidden layer (32–512, with step size 32), and learning rate (sampled in the range $[1 \times 10^{-5}, 1 \times 10^{-2}]$). For the two dense models, ReLU was used as activation function, whereas for the RNN, LSTM, and GRU models, the tanh function was used.

To prevent overfitting, a dropout layer was added after each hidden layer for the dense models, whereas for the simple RNN, LSTM, and GRU models, recurrent dropout were used. For all models, the dropout rate was also tuned (0.0–0.5 with step size 0.1, where a dropout rate of zero is equivalent to no dropout). The simple RNN, LSTM, and GRU models all had a dense output layer, and to regularize this as well, a dropout layer was added before the dense output (where the dropout rate was tuned as described above).

Due to the large number of hyperparameter combinations (making it infeasible to test all within a reasonable amount of time), the Bayesian optimization tuner provided by the Keras library was used [25]. This tuner uses Gaussian processes to select hyperparameters that are likely to improve the model given previous results, and it was assumed that convergence to an optimal set of hyperparameters would be found relatively quickly (maximum number of trials were set to 50). As a last step, the optimal number of epochs were tuned as well, and the final models were re-trained with the best set of hyperparameters (including the best epoch), and finally evaluated on the test sets.

The best set of hyperparameters for each model are given in Table 3.1. For the models that were fit with rolling windows, the length of the input windows can be viewed as a hyperparameter as well, and in Table 3.1, only the window length (indicated in parentheses) for the best performing models are given.

Table 3.1: The best set of hyperparameters for each model. (NA = not applicable.)

Model	Learning rate	Units/dropout, 1st layer	Units/dropout, 2nd layer	Units/dropout, 3rd layer	Units/dropout, 4th layer	Dropout before output layer
Dense model	1×10^{-5}	512 / 0.0	32 / 0.0	512 / 0.3	32 / 0.0	NA
Dense model (6h)	1×10^{-5}	512 / 0.0	-	-	-	NA
RNN model (12h)	1×10^{-5}	32 / 0.0	32 / 0.0	-	-	0.0
LSTM model (12h)	1×10^{-4}	32 / 0.0	32 / 0.0	-	-	0.0
GRU (24h)	1×10^{-5}	32 / 0.5	32 / 0.0	32 / 0.4	256 / 0.5	0.0

4. Results and Discussion

4.1 Model evaluation

Forecasts on the original scale of a time series are easier to interpret, and therefore the transformations made before model fitting were reversed to convert back to μg/m^3 before proceeding with the model evaluation. This required inverting the normalization, and also for the MLR model, using $\exp(\hat{y})$ for the predictions.

4.1.1 Common regression metrics

The RMSE, MAE, and ME for the robust MLR model and the neural network models are shown in Fig. 4.1. Again, for the neural network models that were fit with rolling windows, the length of the input windows are indicated in parenthesis, and only the best performing models (in terms of lowest RMSE) are included here.

As can be inferred from Fig. 4.1, the neural network models all had lower RMSE than the baseline MLR model, with the LSTM model having the lowest value (3.10), closely followed by the dense model (3.18) that was not fit with rolling windows (see Section 3.2.2). Looking at the MAE, also the LSTM model had the lowest value (1.89), again followed by the dense model (1.93), however, for the three remaining neural network models, the MAE were higher than for the baseline MLR model. Though both the RMSE and the MAE measure variability in $e_t(1)$, squaring the errors before averaging (as is done with the MSE) will weight the errors differently than when taking the absolute value. More specifically, the MSE/RMSE penalize large errors more than does the MAE [26]. Given the research question of this thesis, where the emphasis is on pollution peaks (where larger errors are expected), the MSE/RMSE would be the more logical measure to focus on here.

Finally, looking at the ME in Fig. 4.1, the recurrent neural network models all had small ME's, indicating considerably less bias in the forecasts than for the two dense models and the MLR model (which had the largest ME of 0.48). Not only the magnitude, but also the sign of the ME is important, as a positive ME indicates an underestimation bias, whereas a negative ME indicate bias in the opposite direction. Consequently, the robust MLR model clearly gave predictions that were too low, whereas the opposite is true for the two dense models.

4.1.2 Examining the forecast errors

Residual autocorrelations

The Box-Pierce test was used with all models to see if the distribution of the sample ACF's for the forecast errors were approximately normal. As described in Section 2.2.4,

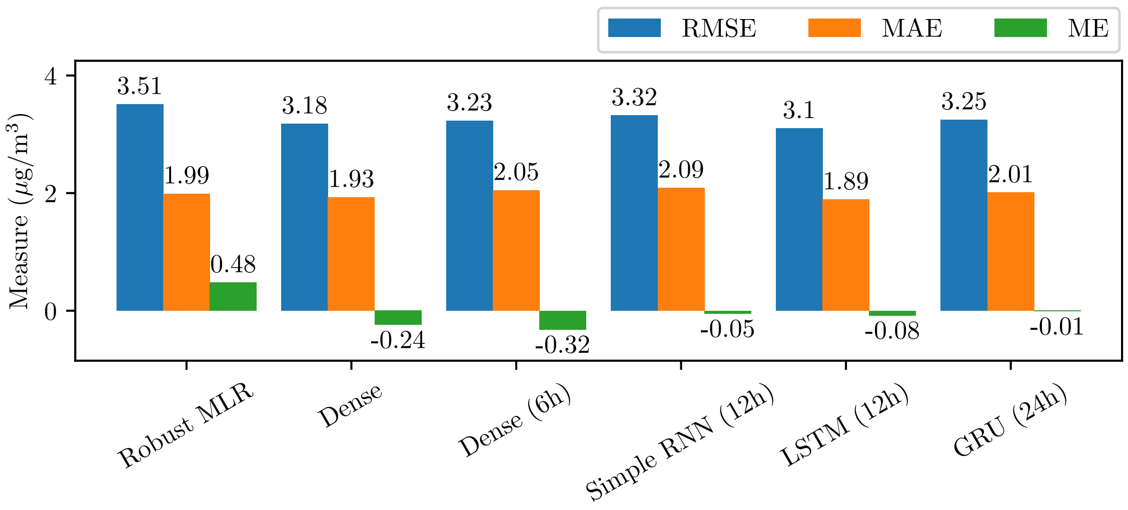

Figure 4.1: Performance metrics for all models.

this is equivalent to testing whether the forecast errors are white noise. Using the 50 first autocorrelations, the Box-Pierce statistic was very large (well above 100) for all models, and the corresponding p-values $\ll 0.001$. The results from the Box-Pierce tests are summarized in Table C.1 in Appendix C. QQ-plots and histogram plots of the forecast errors for all models are also shown in Fig. C.1 and Fig. C.2, respectively, in Appendix C. The errors were not normally distributed for any of the models, as is indicated by the shape of the histograms in Fig. C.2, but also by the departures from the diagonal line in Fig. C.1. Consequently, this suggests that the forecast errors were not Gaussian white noise. Instead, the errors appeared to be strongly correlated and non-random for all models. The structure of the errors are discussed next.

Structure of the forecast errors

A more clear understanding of the structure of the forecast errors can be given by looking at the scatter plots of observed vs. predicted values for each model, shown in Fig. 4.2. The correlation between observed and predicted values are also indicated with the Pearson correlation coefficient (ρ) for each model. If the forecasts were unbiased, the data points would be evenly distributed around the diagonal lines. However, this was not the case for any of the models, as a noticeable tendency to underestimate NO_2 levels in the higher ranges (≥ 40 µg/m^3) can be seen.

The MLR model had the lowest correlation coefficient ($\rho = 0.916$), and also the strongest tendency to underestimate NO_2 values, as indicated by the many data points located below the diagonal line in Fig. 4.2a. Contrary to the neural network models (Fig. 4.2b-f), the MLR model also made a few predictions that were strongly overestimated.

For all neural network models, the correlation coefficient was higher than for the MLR model (and again the LSTM model had the best performance for this metric, $\rho = 0.934$). Furthermore, the predictions were less biased, both in the lower ranges of NO_2 levels, but especially in the higher ranges when compared to the MLR model (which showed clear bias in the predictions already for observations exceeding 25 µg/m^3). However, as pointed out above, all neural network models had the same problem of generating too low predictions at higher NO_2 levels (and this is very noticeable in Fig. 4.2b-f). The ME's for the two dense models indicated a slight overestimation bias (Fig. 4.1). However, from Fig. 4.2b-c, it is clear that this bias only pertained to predictions in the lower ranges, and not the higher ranges where the opposite is true.

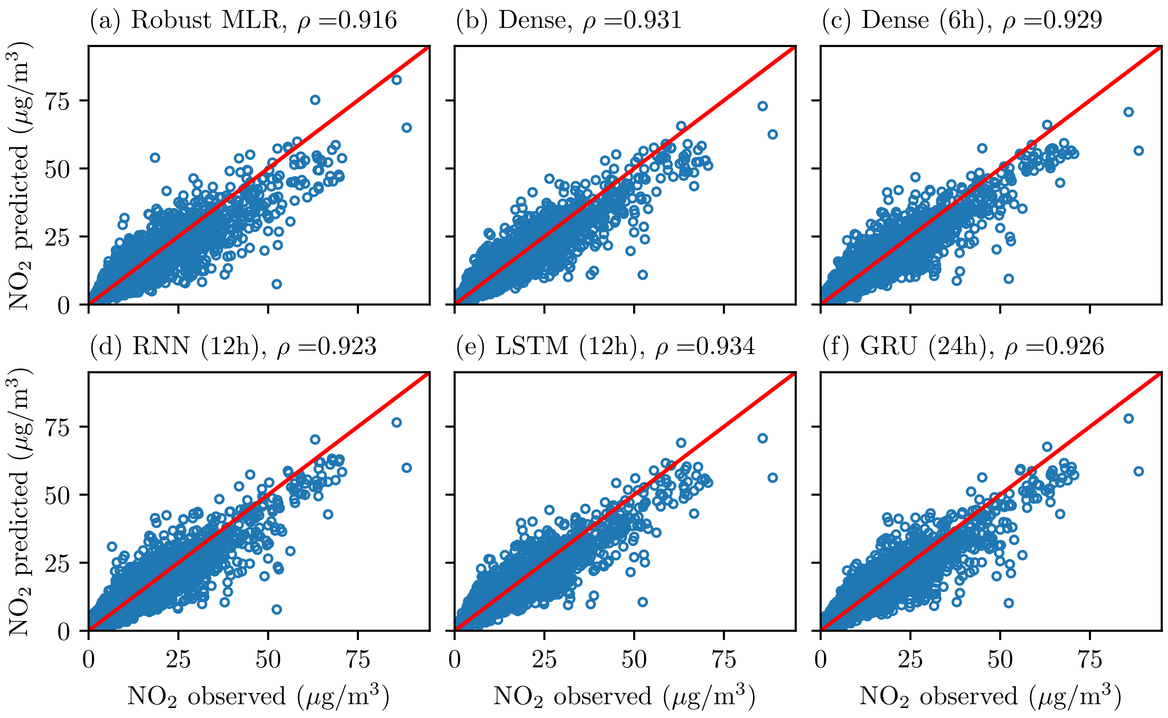

Figure 4.2: Scatterplots of observed vs. predicted NO_2 values.

In Fig. 4.3 on the following page, observed and predicted NO_2 levels during a 10 day period in December 2021 are shown for the robust MLR model (Fig. 4.3a), and the LSTM model (Fig. 4.3b). Only the best performing model is compared to the baseline MLR model in Fig. 4.3, since the forecast errors were similar in character for all models. From these plots, the structure of the forecast errors also become apparent (and merely corroborate the findings related to Fig. 4.2). For example, it can be seen that both models had difficulties predicting the magnitude, as well as duration of, the large NO_2 peak occurring around 7–8 December, though the forecasts of the LSTM model are still better. More accurate forecasts for the LSTM model are also seen during the smaller NO_2 peaks (e.g. the peaks occuring around 3–4 and 6–7 December). At low NO_2 levels however, predictions for both models follow the observed NO_2 values closely.

4.2 Inference for the regression metrics

Due to the fact that the forecast errors for none of the models were Gaussian white noise (and consequently not normally distributed), the Friedman's test was used to see whether there was a significant difference between the models with respect to the MSE. The tests indicated a significant difference ($\chi^2_F(5) = 476.61$, $p \ll 0.001$) between the different models.

Following this, the Bonferroni-Dunn post-hoc test was used for pairwise comparisons. The main finding was that there were significant differences between the baseline MLR model and all neural network models (all p-values were < 0.05). Between the

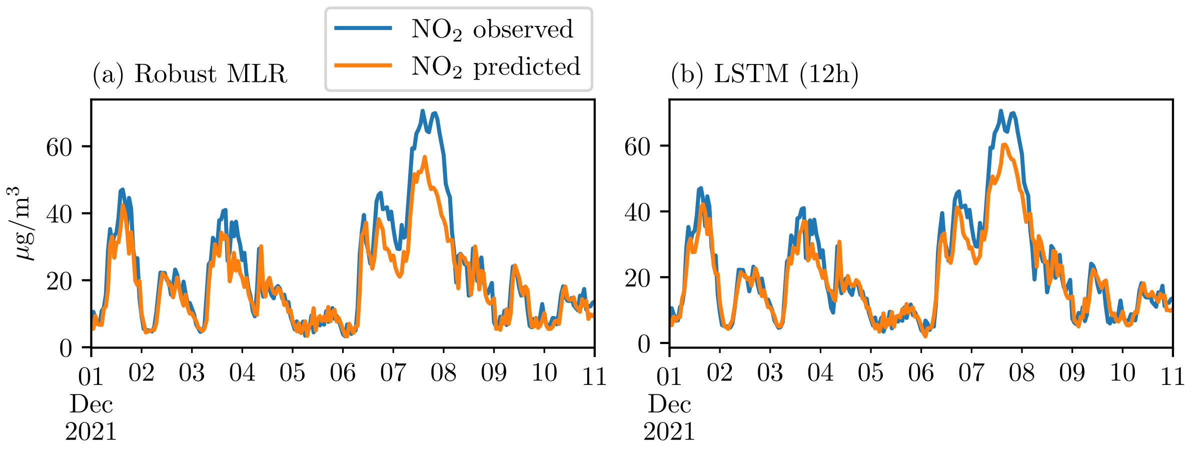

Figure 4.3: Observed and predicted NO$_2$ levels during 10 days in December.

two best performing models however (the LSTM and the dense model), no significant difference was indicated ($p > 0.05$). All p-values for the pairwise comparisons are summarized in Table C.2 in Appendix C.

4.3 Summary and discussion

Common regression metrics indicated that all neural network models generated better forecasts than the baseline MLR model. Moreover, the results were statistically significant, as indicated by Friedman's and the Bonferroni-Dunn tests. However, the improvements were not dramatic, as the best performing neural network model generated forecasts with roughly 12% lower RMSE and about 5% lower MAE compared to the baseline model. (As pointed out above, though not insignificant, less emphasis is put on the MAE here.)

A more noticeable improvement is seen for the ME, for which all RNN models showed superior performance compared to the baseline, as well as the two dense models. Clearly, the RNN models managed to utilize temporal dependencies in the data in a way so to reduce the bias. However, any further advantages for the RNN models are unclear, as there was no significant difference between the best performing dense model and the LSTM model when looking at the MSE. Here, the nature of the time series being modeled is relevant. For example, if temporal dependencies only are important up to a few past time steps (or just one time step) when the next value is to be predicted, a simpler model not explicitly taking the history of the time series into account will produce successful predictions as well. An indication of the importance of previous time steps for the response variable can be given by looking at the magnitude of the coefficients (or the z-statistic) for the robust MLR model (Table B.2 in Appendix B). It can be inferred that the value at lag one dominated in importance, whereas the value at lag two, though still significant, mattered much less. Values of the response variable at lags further back were not utilized in the MLR model, but it can be reasonably assumed that the importance would decrease at every time step going back. Another interesting finding is that for the LSTM model, using the 12h input window gave better results than with the 24h input window. (For the GRU model, the 24h input window gave better predictions, the improvements over the 12h input window were only subtle however). This also suggests that longer-term temporal dependencies were of less importance for predicting NO_2 levels the next hour.

Despite the performance improvements seen for the neural network models (especially the LSTM model), all models failed to successfully capture the structure in the observations, as there was strong autocorrelation in the forecast errors. Judging from Fig. 4.2 and Fig. 4.3, the autocorrelation in the errors are due to the poor predictions made at high NO_2 levels, and especially jerky leaps like the large peak seen around 7–8 of December in Fig. 4.3 are not captured well. Also, the two dense models tended to generate overestimated predictions at lower NO_2 levels, as indicated by the ME's (though this is not clear from Fig. 4.2b-c).

An additional aspect that should be discussed is that of the two model fitting strategies. As explained in Section 3.2, the MLR model and the first dense model were not fit with lagged meteorological variables, and in operational mode these values would need to be replaced with forecasts. Consequently, the performance metrics reported here are under the assumption of meteorological forecasts that are exact. Clearly, this is an unreasonable assumption, and the theoretical performance for these two models are expected to be too optimistic. On the other hand, the models fit with rolling windows only utilize past pollution and weather data, and therefore do not suffer from this added uncertainty. Hence, for these models, the theoretical performance should better reflect the expected performance in operational mode.

5. Conclusions

In this work, several different deep neural network architectures have been explored and compared with a multiple linear regression model for predicting hourly urban background levels of NO_2 using time series data. Generally, the multiple linear regression model was straightforward to implement compared to the deep neural network models, though additional data preparation steps were necessary to ensure that the model assumptions were not violated. The deep neural network models required less data preparation, but took much longer to train and tune.

Across several evaluation metrics, the deep neural network models performed better than the multiple linear regression model, and in particular, a recurrent neural network model (LSTM) consistently had superior performance. The theoretical performance for the best model should also reflect the expected performance in operational mode. Viewing the results in relation to the research question of the thesis (Section 1.3), sudden NO_2 peaks were poorly predicted however, also by the best model, and for all models, forecasts at high NO_2 levels were unsatisfactory. Moreover, none of the models had the desired structure of the forecasts errors, which warrants further model refinements.

Generally when forecasting air pollution, it is common to focus on daily mean levels rather than hourly means [6, 27]. This makes comparisons to other work difficult, but the results herein are similar to that of e.g. Rahimi, 2017 [28], Goulier et al., 2020 [29] and Arsov et al., 2021 [30]. Also SLB-analys' forecasts for the Stockholm metropolitan area are given as risk indexes based on daily means of pollutants [3]. Hourly forecasts though, has the advantage over daily forecasts in that they can indicate during what hours of the day pollution levels could reach hazardous levels, and this knowledge could in some cases be critical. The forecast horizons here are very short-term (one hour), which admittedly limits usability. However, all models in this work could quite easily be adapted to forecast horizons of arbitrary length by simply utilizing the predictions generated. Also, the models could of course be used with other type of air pollutants ($PM_{2.5}$, PM_{10}, O_3, etc.) and in which case, if all are forecast simultaneously, would allow for prediction of an hourly air quality index [2]. Such detailed forecasts could be of great societal value; they could e.g. be used by public health authorities and policy makers. If the forecasts also were made available to the public, they could help with early warning systems aimed at sensitive groups.

Utilizing time series data from multiple air pollution sensors is a challenging task, and this work points to an advantage for the more complex neural networks over the simpler linear model for this type of problem. Further advances in the field of machine learning (specifically deep learning), together with increased availability of massive datasets from multi-sensor air monitoring systems and other relevant data sources (like weather, traffic, and satellite data), offers the potential to cross the conventional limits of forecasting in the near future.

6. Bibliography

[1] World Health Organization, “Ambient air pollution: a global assessment of exposure and burden of disease,” tech. rep., World Health Organization, 2016.

[2] G. W. VanLoon and S. J. Duffy, *Environmental Chemistry.* London, England: Oxford University Press, 3 ed., Sept. 2010.

[3] SLB-analys, “Luften du andas - nu och de kommande dagarna: Utveckling av ett automatiskt prognossystem för luftföroreningar och pollen,” tech. rep., SLB-analys vid miljöförvaltningen i Stockholm, 2021.

[4] M. El-Harbawi, “Air quality modelling, simulation, and computational methods: a review,” *Environmental Reviews*, vol. 21, pp. 149–179, Sept. 2013.

[5] Q. Liao, M. Zhu, L. Wu, X. Pan, X. Tang, and Z. Wang, “Deep learning for air quality forecasts: a review,” *Current Pollution Reports*, vol. 6, pp. 399–409, Sept. 2020.

[6] H. Taheri Shahraiyni and S. Sodoudi, “Statistical modeling approaches for PM10 prediction in urban areas; a review of 21st-century studies,” *Atmosphere*, vol. 7, no. 2, 2016.

[7] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, May 2015.

[8] OECD, *The Economic Consequences of Outdoor Air Pollution.* Paris: OECD Publishing, 2016.

[9] SLB-analys, “Luften i stockholm, Årsrapport 2021,” Tech. Rep. 2022–5787, SLB-analys vid miljöförvaltningen i Stockholm, 2021.

[10] “Luftövervakning.” `https://www.slb.nu/slbanalys/matningar/`. Accessed April 28, 2022.

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning.* Springer New York, 2009.

[12] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to time series analysis and forecasting.* Wiley Series in Probability and Statistics, Nashville, TN: John Wiley & Sons, 2 ed., Apr. 2015.

[13] A. Lindholm, N. Wahlström, F. Lindsten, and T. B. Schön, *Machine Learning: A First Course for Engineers and Scientists.* Cambridge University Press, 2022.

[14] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis.* Wiley Series in Probability and Statistics, Hoboken, NJ: Wiley-Blackwell, 5 ed., Mar. 2012.

[15] J. J. Faraway, *Linear Models with Python.* Chapman & Hall/CRC Texts in Statistical Science, London, England: CRC Press, Dec. 2020.

[16] J. F. Pucer, G. Pirš, and E. Štrumbelj, "A bayesian approach to forecasting daily air-pollutant levels," *Knowledge and Information Systems*, vol. 57, pp. 635–654, Mar. 2018.

[17] F. Chollet, *Deep learning with python.* New York, NY: Manning Publications, Oct. 2017.

[18] D. O. Wackerly, W. Mendenhall, and R. L. Scheaffer, *Mathematical statistics with applications, international edition.* Florence, KY: Brooks/Cole, 7 ed., Oct. 2007.

[19] D. G. Pereira, A. Afonso, and F. M. Medeiros, "Overview of friedman's test and post-hoc analysis," *Communications in Statistics - Simulation and Computation*, vol. 44, pp. 2636–2653, Aug. 2014.

[20] E. Stadlober, S. Hörmann, and B. Pfeiler, "Quality and performance of a PM10 daily forecasting," *Atmospheric Environment*, vol. 42, pp. 1098–1109, Feb. 2008.

[21] SMHI, "Datavärdskap för luftkvalitet." `https://www.smhi.se/data/miljo/luftmiljodata`. Accessed May 3, 2022.

[22] "Environmental monitoring program area: Air." `https://www.naturvardsverket.se/en/environmental-work/environmental-monitoring/environmental-monitoring-program-areas/air/`. Accessed April 27, 2022.

[23] "Historiska data." `https://www.slb.nu/slbanalys/historiska-data-met/`. Accessed April 27, 2022.

[24] A. Gilik, A. S. Ogrenci, and A. Ozmen, "Air quality prediction using CNN+LSTM-based-based hybrid deep learning architecture," *Environmental Science and Pollution Research*, vol. 29, pp. 11920–11938, Sept. 2021.

[25] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi, *et al.*, "Kerastuner." `https://github.com/keras-team/keras-tuner`, 2019.

[26] A. Botchkarev, "Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology," *Interdisciplinary Journal of Information, Knowledge, and Management, 2019, 14, 45-79*, 2018.

[27] S. R. Shams, A. Jahani, S. Kalantary, M. Moeinaddini, and N. Khorasani, "Artificial intelligence accuracy assessment in NO2 concentration forecasting of metropolises air," *Scientific Reports*, vol. 11, Jan. 2021.

[28] A. Rahimi, "Short-term prediction of NO2 and NOx concentrations using multilayer perceptron neural network: a case study of tabriz, iran," *Ecological Processes*, vol. 6, Jan. 2017.

[29] L. Goulier, B. Paas, L. Ehrnsperger, and O. Klemm, “Modelling of urban air pollutant concentrations with artificial neural networks using novel input variables,” *International Journal of Environmental Research and Public Health*, vol. 17, p. 2025, Mar. 2020.

[30] M. Arsov, E. Zdravevski, P. Lameski, R. Corizzo, N. Koteli, S. Gramatikov, K. Mitreski, and V. Trajkovik, “Multi-horizon air pollution forecasting with deep neural networks,” *Sensors*, vol. 21, p. 1235, Feb. 2021.

Appendices

A Monitoring stations

Information about the monitoring stations from which data was used are summarized in Table A.1. The Norr Malma station measure rural-regional background levels. The stations at E4/E20 Lilla Essingen, Hornsgatan 108, and Sveavägen 59 measure urban traffic levels. The Torkel Knutssonsgatan station (the target station) measure urban background levels. As indicated by the coordinates, all stations are relatively close to each other except the Norr Malma station, which is located about 70 km northeast of the Stockholm city centre.

Table A.1: Monitoring stations.

Station	Station code	Longitude	Latitude	Type of monitoring
Norrtälje, Norr Malma	18643	18.631313	59.832382	Rural-Regional Background
Stockholm, E4/E20 Lilla Essingen	18644	18.00439	59.325527	Urban Traffic
Stockholm, Hornsgatan 108	8780	18.04866	59.317223	Urban Traffic
Stockholm, Sveavägen 59 Gata	8779	18.058254	59.340828	Urban Traffic
Stockholm, Torkel Knutssongatan	8781	18.057808	59.316006	Urban background

B Model diagnostics and summary statistics for the multiple linear regression models

Residual plots for the OLS regression are shown in Fig. B.1. From plot (a) and (c), the long-tailed error distribution can be seen, especially in plot (a) where the long tails are indicated by deviations from the straight line. Looking at plot (b), the variance appears stable, and the residuals are scattered in a reasonably random fashion, with no indications of non-linearity. The variance of the residuals also appear stable over time, as indicated in plot (d).

In Table B.1 and B.2, summary statistics are shown for the OLS regression and robust regression, respectively. For the OLS regression (Table B.1), the Durbin-Watson statistic (with a value of 1.943) did not indicate any autocorrelation, and judging from the condition number (52.5), there were no serious issues with multicollinearity among the predictors. The coefficients in the OLS and robust regression models had very similar values (though not identical), and for both models it is clear that the response variable at lag 1 dominated in terms of importance. Also, for both models, the sine wave variable had a high p-value, however, it was not removed as both the sine and cosine waves are needed to model the periodicity.

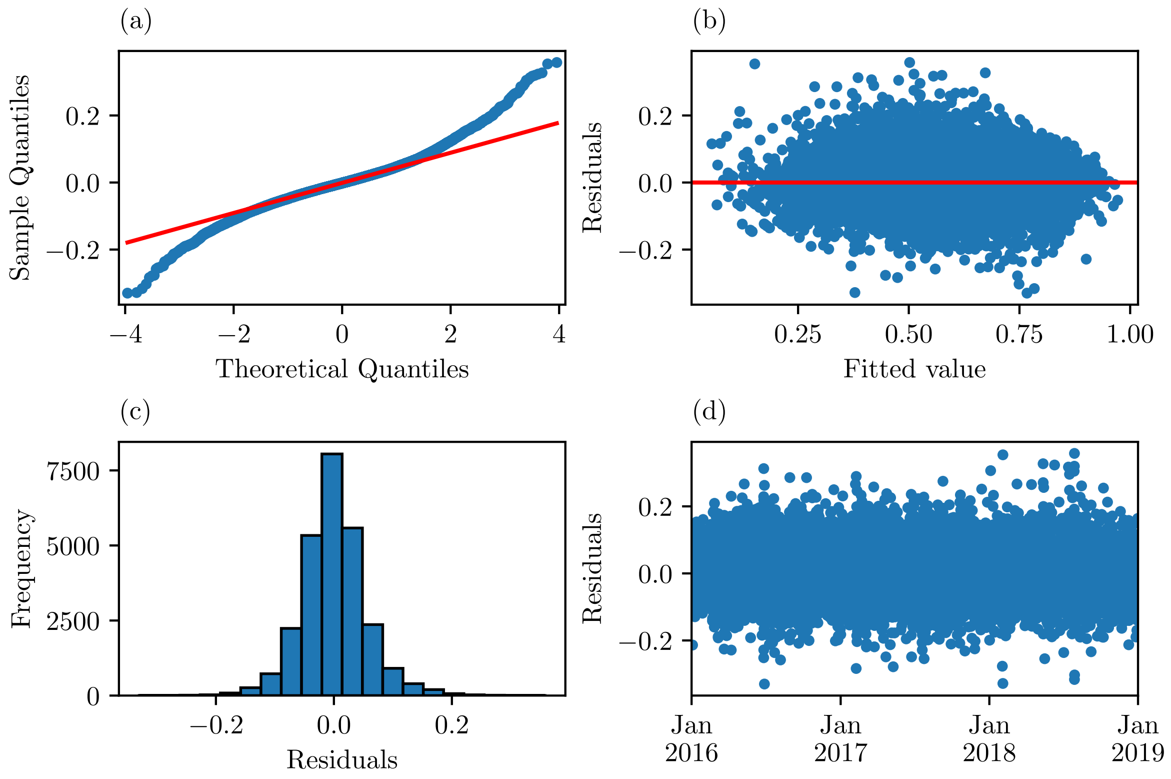

Figure B.1: Residual plots for the OLS regression model.

Dep. Variable:	NO_2, Torkel Knutssonsgatan	**R-squared:**	0.852
Model:	OLS	**Adj. R-squared:**	0.852
Method:	Least Squares	**F-statistic:**	1.381e+04
Date:	Mon, 29 Aug 2022	**Prob (F-statistic):**	0.00
Time:	18:02:49	**Log-Likelihood:**	39098.
No. Observations:	26305	**AIC:**	-7.817e+04
Df Residuals:	26293	**BIC:**	-7.807e+04
Df Model:	11		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
intercept	0.1381	0.005	25.753	0.000	0.128	0.149
NO_2, Stockholm Torkel Knutssonsgatan, lag1	0.9342	0.006	144.333	0.000	0.922	0.947
NO_2, Stockholm Torkel Knutssonsgatan, lag2	-0.1933	0.006	-33.866	0.000	-0.204	-0.182
NO_2, Stockholm Hornsgatan 108 , lag1	0.0420	0.004	11.113	0.000	0.035	0.049
NO_2, Stockholm Sveavägen 59 , lag1	-0.0474	0.004	-11.677	0.000	-0.055	-0.039
NO_2, Stockholm E4/E20 Lilla Essingen, lag1	0.1424	0.005	28.236	0.000	0.132	0.152
Sine day	0.0024	0.001	1.960	0.050	-1.57e-07	0.005
Cosine day	-0.0569	0.001	-40.906	0.000	-0.060	-0.054
Temperature	-0.0079	0.003	-3.058	0.002	-0.013	-0.003
Relative humidity	-0.0201	0.002	-8.296	0.000	-0.025	-0.015
Solar radiation	-0.0970	0.003	-35.575	0.000	-0.102	-0.092
Wind speed	-0.1157	0.003	-33.988	0.000	-0.122	-0.109

Omnibus:	1784.052	**Durbin-Watson:**	1.943
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	6377.932
Skew:	0.282	**Prob(JB):**	0.00
Kurtosis:	5.346	**Cond. No.**	52.5

Table B.1: OLS regression results.

Dep. Variable:	NO_2, Torkel Knutssonsgatan	**No. Observations:**	26305
Model:	RLM	**Df Residuals:**	26293
Method:	IRLS	**Df Model:**	11
Norm:	HuberT		
Scale Est.:	mad		
Cov Type:	H1		
Date:	Mon, 29 Aug 2022		
Time:	18:03:09		
No. Iterations:	26		

	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
intercept	0.1276	0.005	26.660	0.000	0.118	0.137
NO_2, Stockholm Torkel Knutssonsgatan, lag1	0.9688	0.006	167.776	0.000	0.957	0.980
NO_2, Stockholm Torkel Knutssonsgatan, lag2	-0.1999	0.005	-39.270	0.000	-0.210	-0.190
NO_2, Stockholm Hornsgatan 108 , lag1	0.0330	0.003	9.802	0.000	0.026	0.040
NO_2, Stockholm Sveavägen 59 , lag1	-0.0379	0.004	-10.456	0.000	-0.045	-0.031
NO_2, Stockholm E4/E20 Lilla Essingen, lag1	0.1246	0.004	27.702	0.000	0.116	0.133
Sine day	0.0008	0.001	0.687	0.492	-0.001	0.003
Cosine day	-0.0522	0.001	-42.106	0.000	-0.055	-0.050
Temperature	-0.0071	0.002	-3.086	0.002	-0.012	-0.003
Relative humidity	-0.0200	0.002	-9.268	0.000	-0.024	-0.016
Solar radiation	-0.0900	0.002	-36.969	0.000	-0.095	-0.085
Wind speed	-0.1045	0.003	-34.420	0.000	-0.110	-0.099

Table B.2: Robust linear model regression results.

C Histogram and qq-plots of the forecast errors, Box-Pierce test results, and results from the Dunn's test

The results from the Box-Pierce tests for all models are summarized in Table C.1 below.

Table C.1: Results from the Box-Pierce tests.

Model	Q_{BP}	p-value
Robust MLR model	1520.07	$\ll 0.001$
Dense model	866.76	$\ll 0.001$
Dense model (6h)	607.94	$\ll 0.001$
Simple RNN model (12h)	238.62	$\ll 0.001$
LSTM model (12h)	315.34	$\ll 0.001$
GRU model (24h)	506.06	$\ll 0.001$

The p-values from the Bonferroni-Dunn post-hoc tests for the MSE are given in Table C.2.

Table C.2: p-values for the pairwise comparisons from the Bonferroni-Dunn post-hoc test for the MSE.

Model	Robust MLR	Dense	Dense (6h)	Simple RNN (12h)	LSTM (12h)	GRU (24h)
Robust MLR	1.0	0.00665	$\ll 0.001$	$\ll 0.001$	0.00170	$\ll 0.001$
Dense	0.00665	1.0	$\ll 0.001$	$\ll 0.001$	1.0	$\ll 0.001$
Dense (6h)	$\ll 0.001$	$\ll 0.001$	1.0	1.0	$\ll 0.001$	0.04855
Simple RNN (12h)	$\ll 0.001$	$\ll 0.001$	1.0	1.0	$\ll 0.001$	0.00044
LSTM (12h)	0.00170	1.0	$\ll 0.001$	$\ll 0.001$	1.0	0.00001
GRU (24h)	$\ll 0.001$	$\ll 0.001$	0.04855	0.00044	0.00001	1.0

Quantile-quantile plots and histogram plots of the forecast errors are shown, respectively, in Fig. C.1 and Fig. C.2 below. It can be inferred from these plots that none of the models generated forecasts with the desired structure of the errors (i.e., Gaussian white noise).

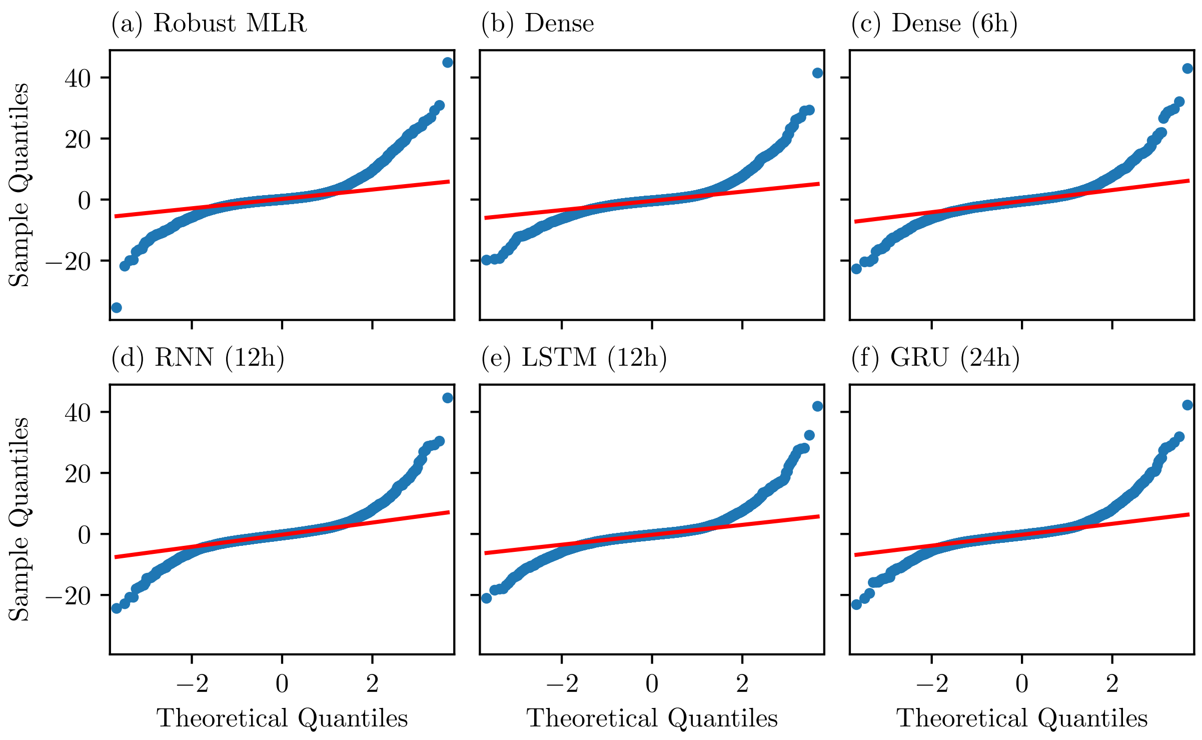

Figure C.1: Normal probability plots for the forecast errors.

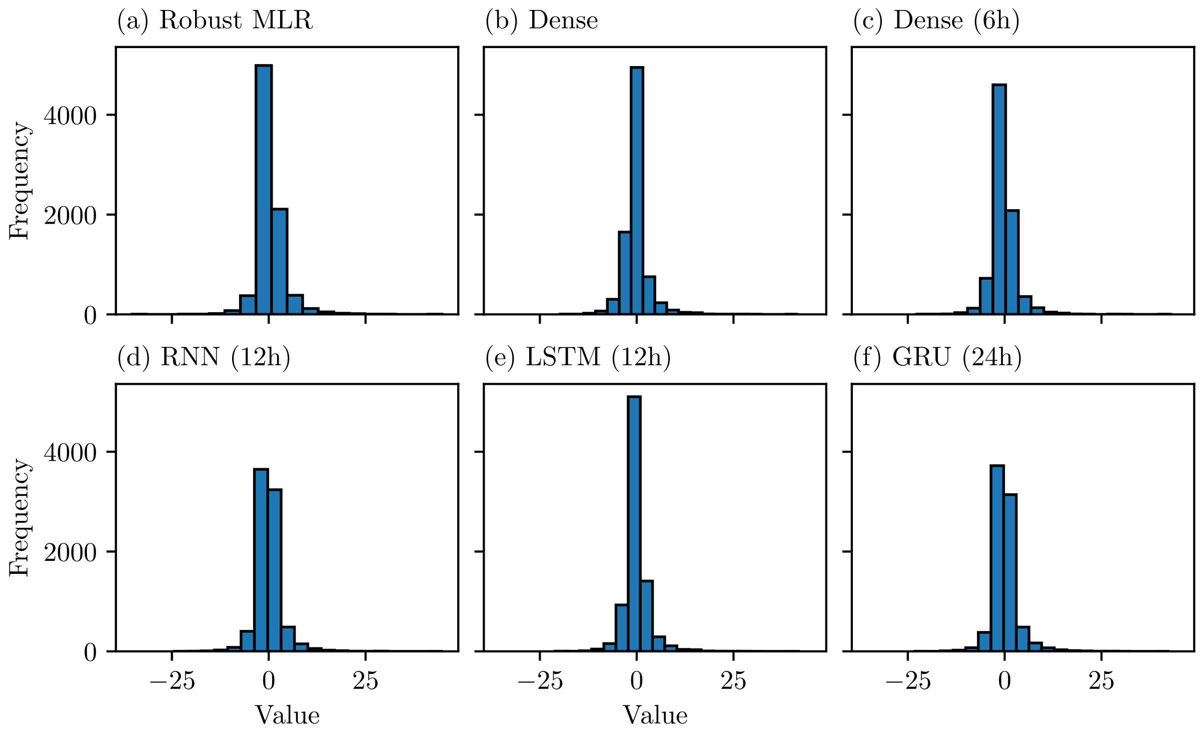

Figure C.2: Histogram for the forecast errors.