A 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

Stockholm
University

Abstract

Synopsis

Background

Problem

Research Question

Method

Result

Discussion

Acknowledgements

Contents

List of Figures

List of Tables

List of Abbreviations

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 have shown promising results with air quality forecasts, especially deep neural networks ”tailored” to sequence and time series data [5,6]. In this work, a few deep learning architectures as well as more straight-forward linear models 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

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 [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 regulatory 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].

NO_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 [13]. Generally in regression, the weights are learned by minimizing a cost function

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

where $\hat{\boldsymbol{\beta}}$ is the vector of estimated model parameters ($\hat{\beta}_0$, $\hat{\beta}_1, ...,$ $\hat{\beta}_n$), $\hat{y}_i$ is a prediction and y_i is a training data value [13]. In Eq. (2.2.1) the squared error loss is used as loss function, and the cost is simply the loss averaged over the training data.[1] Depending on the model, minimizing $J(\hat{\boldsymbol{\beta}})$ 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

[1] How cost and loss functions are defined can vary slightly in the literature, but in this work, the same definitions as in Lindholm et al. [13] are adopted.

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. (3.2.1) 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 frequent extreme values in the data, in which case so-called robust estimation techniques are more appropriate than OLS [14].

M-estimators is a class of robust estimators where a modified version of Eq. (2.2.1) is used 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

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

where t is a robust estimate of σ [15]. Minimizing Eq. (2.2.4) is done iteratively by solving weighted versions of the least squares normal equations and recomputing the weights until convergence is reached [14]. In effect, this results in less importance being put on 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 widely used diagnostic for multicollinearity is the condition number, where a value below 100 is preferred [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]. Non-linear regression models, better suited to capture complex input-output relationships, have also found extensive use in air pollution forecasts, and in the next section, some of these models are reviewed.

2.2.3 Non-linear regression models and deep neural networks

2.2.4 Evaluating regression models

Common evaluation metrics

Classifying prediction results

2.3 Summary and motivation for this work

3. Methodology

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 [16]. 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 [17]. 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 [16].

Monitoring stations

In Stockholm County, there are 19 stationary sites for air pollution monitoring [10], and initially, data from each site was considered. However, not all stations measure hourly NO_2, and for some stations the data were irregular. Therefore, data from four sites with hourly NO_2 measurements (in µg/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 had to be excluded.) For the station at which NO_2 predictions subsequently were to be made (Torkel Knutssonsgatan), hourly meteorological data was also utilized. More specifically, these meteorological variables were temperature (in °C), precipitation (mm), atmospheric pressure (hPa), relative humidity (as %), solar radiation (W/m^2), and wind speed (m/s). The meteorological data were downloaded from SLB-analys' webpage [18].

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) [16]. 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.

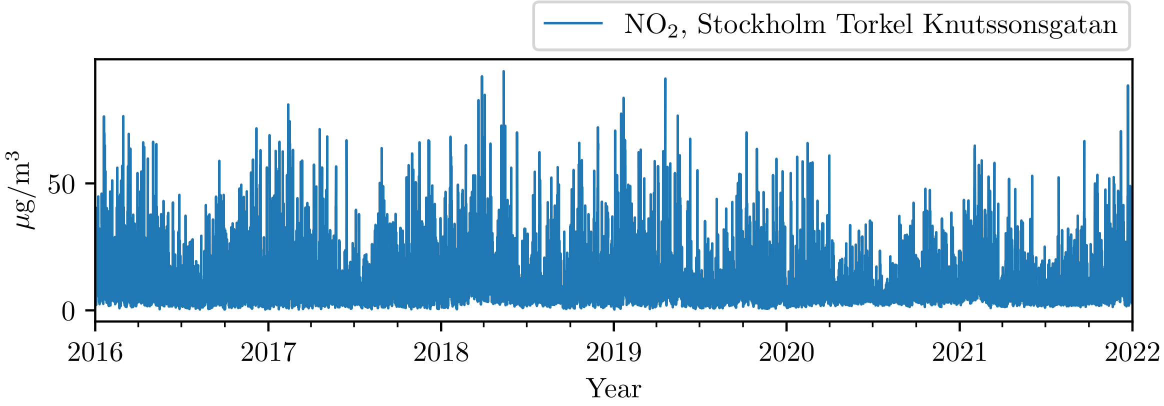

Figure 3.1: NO_2 data for Torkel Knutssonsgatan.

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

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

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

where timestamp is in UNIX epoch time[1] (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 on the next page.

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”

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

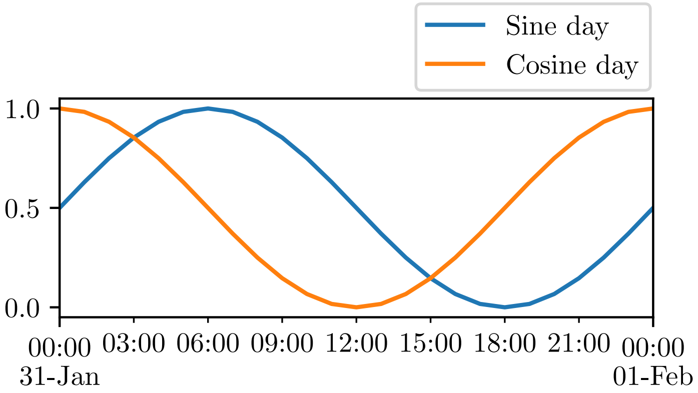

Figure 3.2: Temporal variables for day as sine and cosine waves.

[19]. 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.

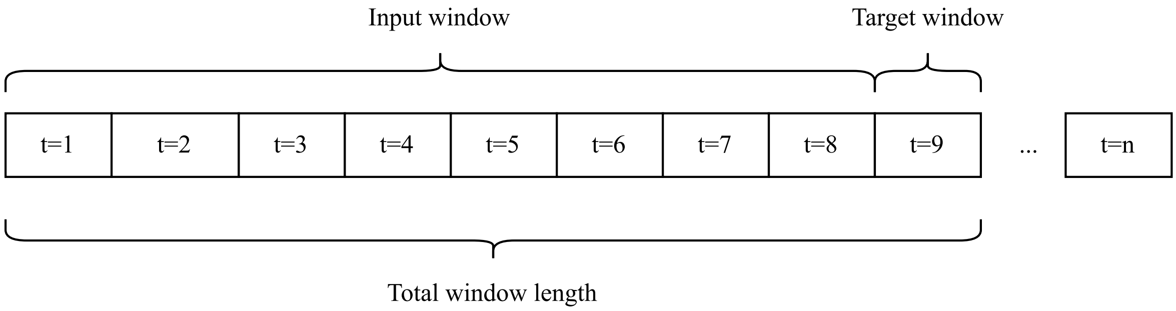

Figure 3.3: Rolling window approach for time-series data.

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 an unusual year with regards to 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 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-estimates (and Huber's 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. From Fig. 3.4, it is evident that the least important predictors brought essentially no improvements to the model, and

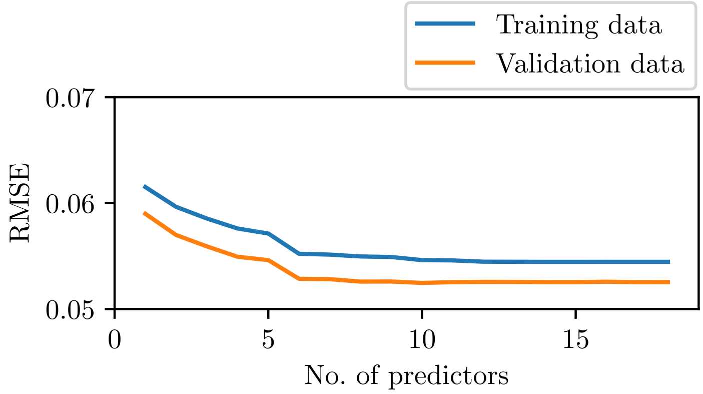

Figure 3.4: RMSE for each candidate model generated by RFE.

they were therefore removed. More specifically, these variables were; precipitation, atmospheric pressure, the Norr Malma NO_2 data, 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 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 learning models

Initially for all deep learning models, the hyperparameters tuned were; number of layers (1–5), number of units in each layer (32–512, with step size 32), and learning rate (sampled in the range $[1 \cdot 10^{-5}, 1 \cdot 10^{-2}]$). If a model showed signs of overfitting (checked by inspecting the learning curves during training), a drop-out layer was added as a last layer, where the dropout rate was also tuned (0–0.3, with step size 0.05). Moreover, input windows of different lengths (12 h and 24 h) were tested as well.

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 [21]. 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. Furthermore, the same input variables as for the final MLR model were used as this provides a more direct comparison of the performance between the two type of models (i.e., the MLR model and the deep neural networks).

As a last step, the optimal number of epochs were tuned as well, and all models were re-trained with the best set of hyperparameters (including the best epoch), and finally evaluated on the test set. The final hyperparameters for each model are summarized in Table 3.1.

Table 3.1: Hyperparameters for the best performing models.

	No. of layers	No. of units per layer	Learning rate	Dropout rate
Dense model (12 h)	4	256/256/256/256	$1 \cdot 10^{-5}$	0.2
RNN model (12 h)	3	126/160/32	$6.79 \cdot 10^{-3}$	-
LSTM model (12 h)	2	32/160	$4.69 \cdot 10^{-5}$	0.2

4. Results and Discussion

4.1 Regression metrics

Forecasts on the original scale of a time series (in this case μg/m^3) are easier to interpret, and therefore the transformations made before model fitting were reversed. This required inverting the normalization, and also in the case of the MLR model, using $\exp(\hat{y})$ for the predictions. For all deep learning architectures, the 12 h input window gave better predictions than with the 24 h input window, and in Table 4.1, common performance metrics for the best performing deep learning models are summarized together with the corresponding performance for the MLR model.

From Table 4.1, it can be inferred that the dense model had the best performance in terms of MSE/RMSE, closely followed by the LSTM model. The RNN model, however, only had slightly better performance than the (baseline) MLR model. Interestingly, the MAPE follows an entirely different pattern, where the MLR model had the lowest value (20%), and the LSTM model the highest (29.6%). This indicates a stronger bias in the predictions by the deep learning models compared to the MLR model.

Table 4.1: Performance metrics for the MLR and deep learning models.

Model	MSE	RMSE	MAPE
Robust MLR model	12.341	3.513	0.20
Dense model	10.621	3.259	0.243
RNN model	12.174	3.489	0.265
LSTM model	11.0	3.317	0.296

In Fig. 4.1, where observed vs. predicted NO_2 levels are shown for each model together with the corresponding correlation coefficient, the bias in the predictions for the deep learning models can be seen. For example, looking at plots (b)–(d), there is a consistent pattern of too high predictions (indicated by the unequal distribution of observations around the red line). For the MLR model (Fig. 4.1a), though less biased, the bias goes in the opposite direction, with a tendency to make too low predictions. The correlation coefficients followed the same pattern as MSE/RMSE, with the dense model having the highest ρ.

In Fig. 4.2, predicted and observed NO_2 levels for all models during a 9 day window (1st to 9th of Dec) from the test set are shown. Comparing the MLR model with the deep learning models in Fig. 4.2, it is clear that the MLR model performed worse during pollution peaks (e.g. the peak observed around 7–8 of Dec was not predicted very accurately). The model best predicting peaks during this time window appears to be the RNN model, but again the RNN model is clearly biased and also had the lowest ρ among the deep learning models.

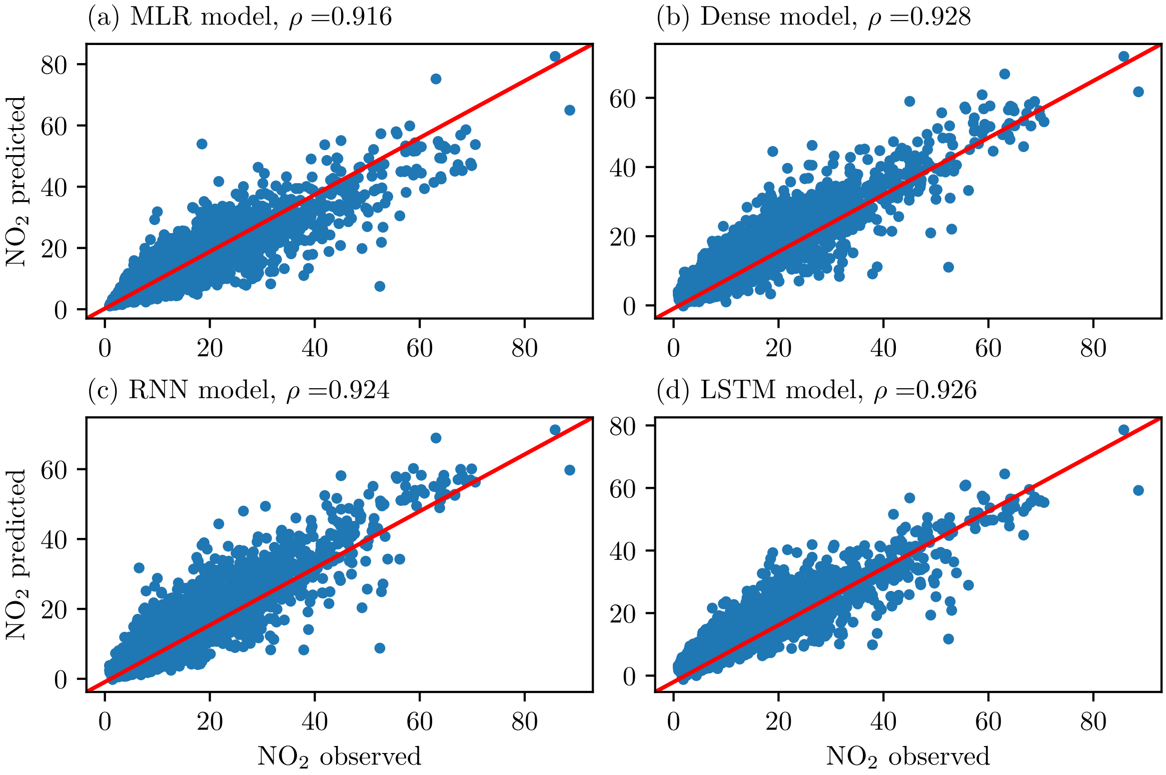

Figure 4.1: Actual vs. predicted NO_2 levels.

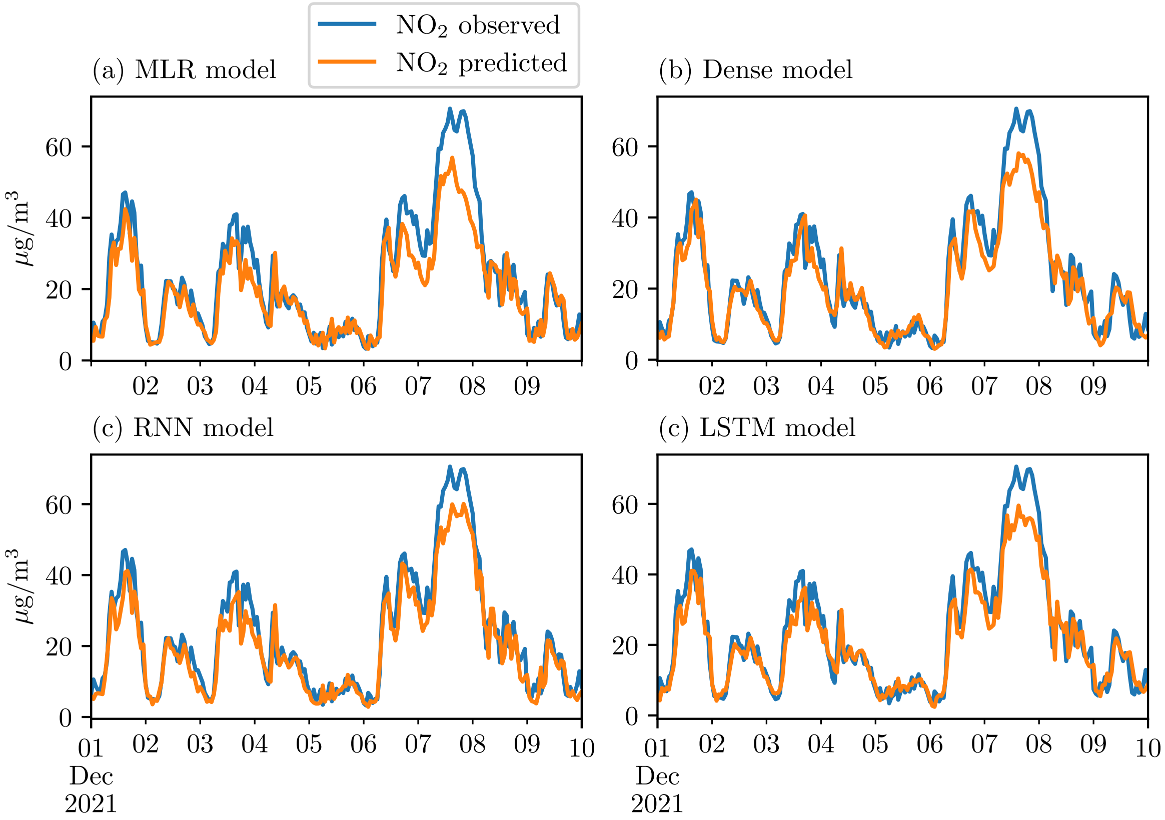

Figure 4.2: Actual and predicted NO_2 for 10 days in Dec.

4.2 Classification of predictions

5. Conclusions

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

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

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

[19] 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.

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

Appendices

A Monitoring stations

Information about the monitoring stations from which data was used is summarized in Table A.1.

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.

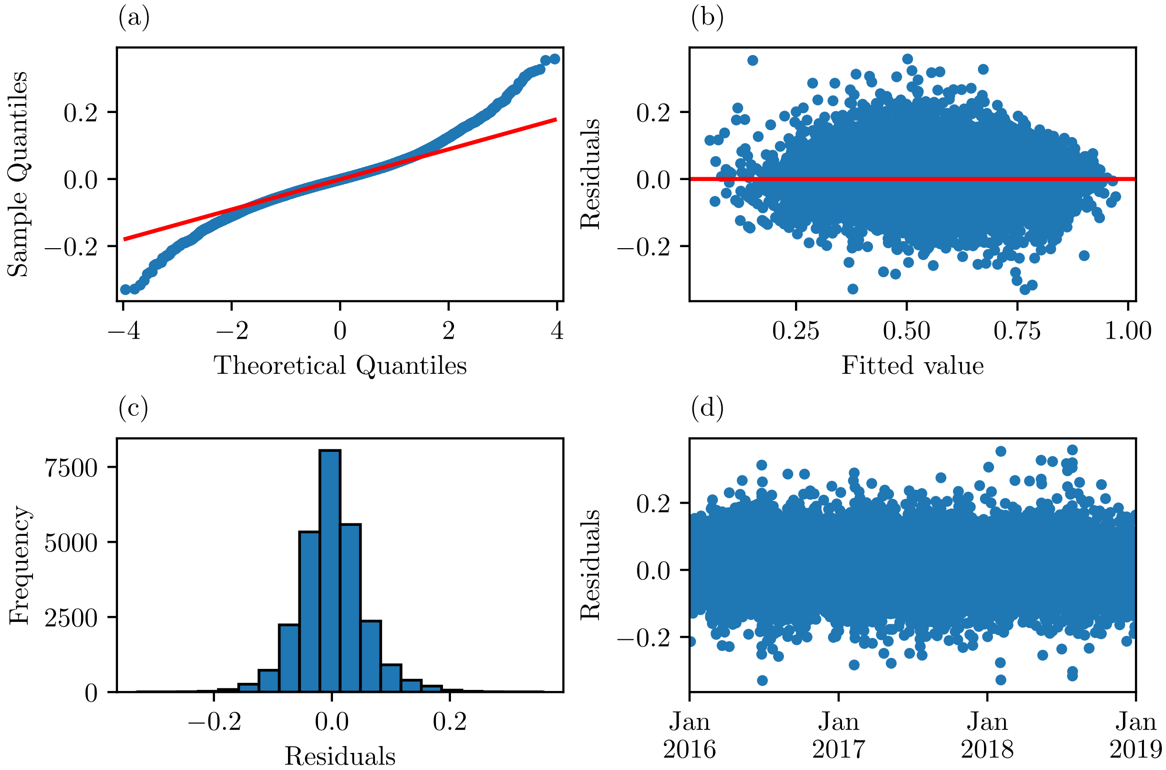

Figure B.1: Residual plots for the OLS regression model.

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.