

# NO<sub>2</sub> Forecasting using Sentinel 5P and Weather Forecast data: a Deep Learning Approach

Huriel Reichel

## Introduction

Nitrogen dioxide ( $NO_2$ ) is a greenhouse gas strongly related to respiratory diseases (Heinrich et al., 1999, Hagen et al., 2000) often used as reference for pollution indicators together with carbon dioxide ( $CO_2$ ). Monitoring the presence of this gas in the atmosphere is therefore of high health importance, and the same could be said of its forecasting.

Classic weather (including air quality) forecasting makes use of a series of data-sets, including weather stations, satellite imagery, radar data, etc., and includes them for computation in physical models requiring super computers ((Rasp et al. 2020)). However, other strategies have been investigated that demand less computational cost and can reach similar results (Rasp and Thuerey (2021)). These strategies involve using deep learning for the predictive analysis, mainly using time-series data from weather stations (Kurt et al. (2008); Russo and Soares (2013); Tsai, Zeng, and Chang (2018); Kaselimi et al. (2020); Masood and Ahmad (2021); Samal et al. (2021); Heydari et al. (2021)).

By considering deep learning as a predictive model for pollution, one cannot ignore its ability to detect multiple variability and abstractness in data. Not only that, but it can take into account different sources of data. And when considering the availability of data related to  $NO_2$  monitoring, one cannot ignore the imagery from Copernicus's SENTINEL 5P (S5P) satellite. This spacecraft has a daily temporal resolution by covering the entire earth, which makes of it a substantial possibility for a world-wide forecasting scenario, especially when using deep learning. Making use of this dataset and deep learning's abstractness capabilities, it is possible to forecast the whole pollution surface (pixel by pixel) in different future times, in spite of considering the aggregated time-series singularly.

As a primary aim, a deep learning model is trained and evaluated using beyond time-series data, but the series of satellite imagery from SENTINEL 5P for  $NO_2$  daily forecasting. The architecture to be applied is inspired by the one used by Shi et al., 2015, Liu et al., 2018 and

Wen et al., 2019, and consists of a Convolutional Long Short-Term Neural Network (CNN-LSTM). The main rationale is to make use of both space (through the convolution - CNN) and time (through the LSTM) to forecast new “frames” in SENTINEL 5P images. This has already been tested in similar approaches, as the one from Heydari et al. (2021 Heydari et al. (2021)), although what has not been explored yet is the use of weather related covariables that could aid the forecasting goal. Many weather related phenomena affect the chemical reactions behind  $NO_2$  formation in the atmosphere, which are also variables measured spatially and temporally. The goal is then to compare a CNN-LSTM model with weather related covariables and without them in terms of computational complexity and mostly accuracy in daily forecasts.

Nevertheless, before one reaches that, the next topic is here to present to give some deeper background on the topic of pollution forecasts and Deep Learning for the purpose of forecasting it.

## Background

### Weather and Pollution Forecasting

As before mentioned,  $NO_2$  is strongly associated with respiratory diseases (Zhang, Wang, and Lu (2018); Jo et al. (2021)), hence it is deeply studied and often considered a target variable in pollution oriented research. This is only more clear, when looking at works focused on forecasting pollution (Shams et al. (2021) ; Pawul and Śliwka (2016) ; Ebrahimi Ghadi, Qaderi, and Babanezhad (2018) ; Mohammadi et al. (2016) ), where regardless of the model used,  $NO_2$ , among other possible chemicals, is central in these modelling works. In that sense, it is of high importance to first better understand this chemical, and mostly how it is formed into the atmosphere.

Nitrogen dioxide ( $NO_2$ ) is a gaseous air pollutant that forms in the atmosphere primarily through the oxidation of nitrogen oxide ( $NO$ ) in the presence of sunlight and other atmospheric constituents. The formation of  $NO_2$  is closely linked to several chemical and physical processes, and weather conditions play a significant role in its formation, transformation, and dispersion in the atmosphere.

To what concerns the formation mechanism of this chemical, sunlight is crucial for the so called process of photochemical conversion of  $NO$  into  $NO_2$  when combined with  $O_3$  (Ozone) (Wood et al. (2009)). This is the case even considering that anthropogenic sources like vehicle emissions, industrial activities and power generation. Hence sunlight availability is of root importance for the presence of  $NO_2$  in the atmosphere. Not only that, temperature acts as a catalyst for chemical reactions, and usually the higher, the higher the chance for  $NO_2$  presence (Wood et al. (2009) ). Other factor, just as important are phenomena directly linked to the atmosphere’s stability, causing the movement of the gas among the different air layers. Those variables can be indicated by measures of wind speed, wind direction and precipitation, for example. Grundström et al (2015 - Grundström et al. (2015) ) appoints wind speed,

vertical temperature gradient and weather type as variables with a strong relationship with the presence of  $NO_2$  .

A similar work is presented by Samal et al (2021 Samal et al. (2021) ), which compares the use of several different models to forecast PM2.5, another pollutant indicator, making use of weather related variables as supporters, being them wind speed, temperature, wind direction, rainfall, and humidity. In their work, this relationship is tested in those models and it is left as clear how important they are and that weather variables should be used for more accurate pollution forecasting. In summary, one could add that weather is extremely relevant for explaining pollution and it should be considered when attempting to forecast it in a data-driven approach.

### Time Series Forecasting

Pollution forecasting is from many perspectives a time-series challenge (Freeman et al. (2018); Samal et al. (2021) ) and it has been approached as so by researchers. A time-series is essentially a set of observations stored in a sequence, which finally corresponds to time Chatfield (2000). Regardless of the model, if within a data-driven approach, the logic is to use past information to take assumption on the future. Any new information will be given based on past one. This is the approach present in the works of Zhu et al. (2018) and Kumar and Jain (2009) that make use of ARIMA based models, but also of works that developed neural networks for similar tasks (Freeman et al. 2018; Agirre-Basurko, Ibarra-Berastegi, and Madariaga 2006; Kurt et al. 2008; Samal et al. 2021; Tsai, Zeng, and Chang 2018).

In general, given the abstraction level of the phenomenon, deep learning approaches tend to reach more accurate results (Samal et al. (2021) ; Pawul and Śliwka (2016); Shams et al. (2021); Rasp et al. (2020) ) than standard statistical or even other machine learning approaches (“shallow” ones). Nevertheless, the limit of what be done with deep learning models is still far to be reached, and research reaching more and more accurate results are constant. Exploring this further is therefore a systemic scientific work.

Moreover, Wen et al. (2019) makes use of a sophisticated neural network architecture for the forecast of pollutants, whilst Samal et al. (2021) goes further and uses a similar model to forecast the same variable, although now using weather supportive variables. Although, one important aspect is missing in all of these works. The analysis of the spatial pattern of results. Space is completely ignored in the analysis of results, and every model, pehaps even includes the spatial component in the modelling itself, but aggregates it to a typical times series in the end. Questions related to the spatial pattern of the forecast errors or the presence of spatial dependence in forecasts are all neglected when the topic of pollution forecasting is dismantled into having only the temporal component.

Whenever decision have to be made regarding pollution, this decision has boundaries associated to political frontiers. When considering the temporal component only, space and important aspects of it are neglected in the analysis. Hence, whenever research has been made about the

use of deep learning for pollution forecasting, it is either space which is forgotten in the model or in the results, or weather that is not being considered as a covariable. A gap in literature is therefore present.

## Spatial Data Cubes and Sentinel 5P

Once more, it was mentioned that past work is above all focused on “time-only” models. The input data for these models is based overall on sensors on-board weather stations spread all over Earth’s surface or even other systems constantly measuring the desired variable. For that reason, if these stations are not considered into a single model, space is ignored. Nevertheless, now considering a space oriented model, what is the kind of input data one can expect?

Fortunately, Earth Observation technology has been substantially advancing and the deployment of satellites in closer ranges are constantly imaging Earth’s and other planets surfaces for full coverage data. These includes RGB imagery, Infrared imagery, Water Coverage, Albedo, Digital Elevation data, and many others. Among the several missions to do so, an important spotlight shall be given to the Copernicus Missions for the European Space Agency (ESA). The completely free-access terabytes of data derive mainly from the Sentinel satellites aimed at giving Europe autonomous capacity in plenty of Earth Observation tasks (Jutz and Milagro-Pérez (2020)). For the task of pollution forecasting, there is Sentinel 5P (S5P), which on-boards the [TROPOMI](#) sensor, the most advanced multispectral imaging spectrometer to date (2023). With its global coverage, TROPOMI measures the ultraviolet and visible, near-infrared and shortwave infrared spectral bands, what supports a high accurate capture of pollutants such as  $NO_2$ ,  $O_3$ ,  $CH_2O$ ,  $SO_2$ ,  $CH_4$  and  $CO$ .

To what refers to weather related data, considering the same format, *i.e.*, preserving the spatial component, an important source of that is the European Centre for Medium-Range Weather Forecasts (ECMWF). Among their many datasets, a very popular one is ERA5 (Reanalysis v5), which has also global coverage and ranges from today to 1940. It contains several weather related data which can all be freely accessed through an API or a web-portal (Copernicus Climate Change Service (2023)).

This kind of dataset, where both space and time are covered could be referred as the so called spatial data cubes (Lu, Appel, and Pebesma (2018)). The idea is that array data with  $n$  dimensions are associated with geospatial coordinates and a time reference. Figure 2 below from Pebesma and Bivand (2023) represents the data cube having  $x$  and  $y$  coordinates (longitude and latitude, for instance) and also the time in a  $z$  axis. The data itself, whatever it refers to, represents the values itself that the array will possess. Therefore, the spatial and temporal information is to some extent masked by the array dimensions. Moreover, It is important to mention that the cube will then have even more dimensions considering that there could be covariables, such as the weather related ones discussed before. This will of course add complexity, but the most interesting part is that this is still useful.



Figure 1: Illustrative Figure of the S5P satellite and the variables its sensor (TROPOMI) is constantly measuring. Source: ESA

The structure of the data cube is especially interesting for the purpose of forecasting atmospheric pollutants as it can be quite simply converted into a tensor, the input format required for a neural network in most applications. The topic of a tensor will be discussed further.

### Spacetime Deep Learning

Given the high dimensionality and hence high complexity of the input data coming from spatial data cubes, one may be asking how can this level of level of multidimensionality be treated and maintained with any model? This is possible within a deep learning framework, making use of deep neural networks.

Neural networks are computational models inspired by the structure and function of biological neural networks in the human brain (Wei et al. (2020)). They consist of interconnected layers of artificial neurons, also known as nodes or units. Each neuron takes inputs, performs a computation, and produces an output that is transmitted to other neurons. Through a process called “training”, neural networks learn to adjust the strengths of connections (weights) between neurons in order to approximate desired input-output mappings. Figure 3 demonstrates the neurons and weights that programmatically iterated to reach optimal results.

The exercise present in Figure 3 is that of an image classification. An input image of a dog is given as input array and it be processes by the different layers, which nodes will be giving

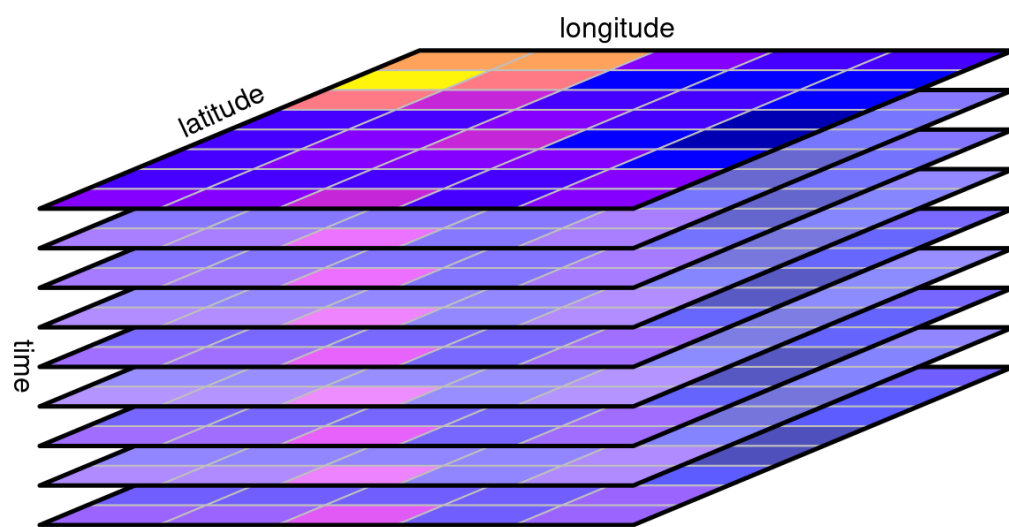


Figure 2: Representation of a Spatial Data Cube. Source. Pebesma and Bivand, 2023

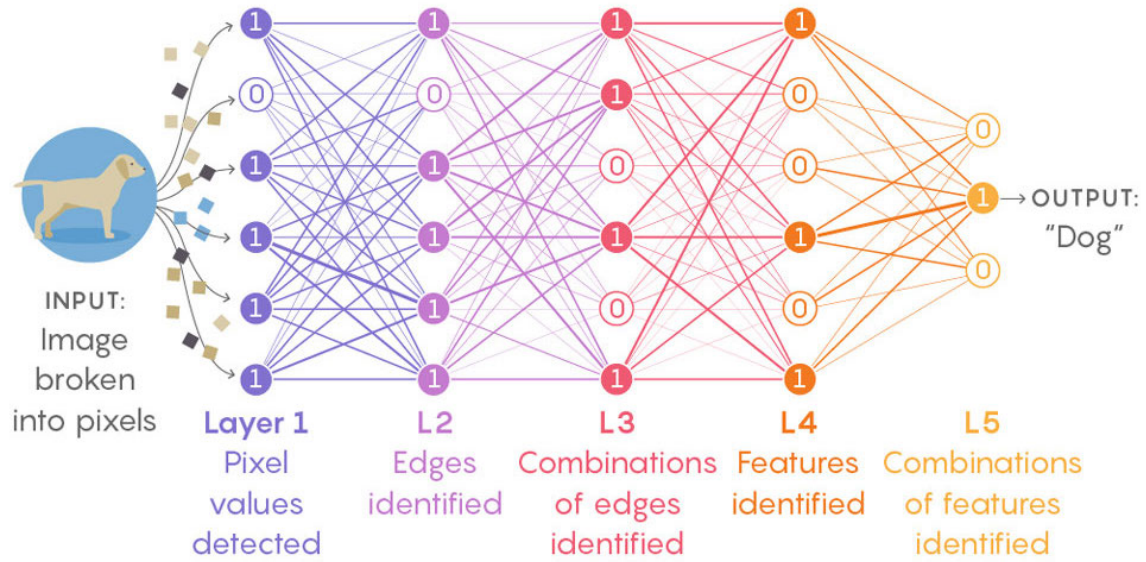


Figure 3: Neural Network model, represented with the objective of image classification. Source: Wolchover, Natalie (2019)

iterative weights until the array is correctly assigned the desired label. Of course, this requires several inputs to be reachable, hence being a considerable extensive methodology.

The key advantage of neural networks in handling high-dimensional data lies in their capacity to learn intricate patterns and relationships within the data, capturing intrinsic abstractness in it, as seen in (Agirre-Basurko, Ibarra-Berastegi, and Madariaga 2006; Masood and Ahmad 2021; Pawul and Śliwka 2016; Zhao et al. 2019). By utilizing deep neural networks with multiple hidden layers, known as deep learning, these models can automatically extract hierarchical representations of features from the input data. This hierarchical representation allows neural networks to effectively capture complex dependencies and interactions between different dimensions, such as spatial, temporal, and covariable aspects of a spatial data cube.

Here is where tensors come into play. A tensor is a mathematical object that can represent multi-dimensional arrays of data Panagakakis et al. (2021). In the context of neural networks and spatial data cubes, tensors provide a natural framework for organising and manipulating such high-dimensional data. A spatial data cube can be conceptualized as a tensor, with each dimension corresponding to a specific aspect of the data, such as spatial coordinates, time, and covariables. The elements of the tensor store the actual values of the data at each point in the cube. Neural networks process tensors as inputs, perform computations on them through various layers, and produce output tensors. The network's parameters, including weights and biases, are learned through training to optimize the model's performance on a given task, such

as prediction or classification. A deeper notion of how data cubes can be interpreted as tensors is present in Mahoney, Maggioni, and Drineas (2006) .

To handle the computational demands of working with large-scale spatial data cubes, neural networks often require significant computational resources. Graphics Processing Units (GPUs) play a crucial role in accelerating neural network computations. GPUs excel at parallel processing, which allows them to perform many calculations simultaneously. This parallelism enables efficient processing of large amounts of data, making GPUs well-suited for training and inference tasks involved in handling big data.

In summary, neural networks are highly effective in handling high-dimensional spatial data cubes, thanks to their ability to learn complex patterns and relationships. Tensors provide a natural representation for organizing and manipulating such data, with each dimension corresponding to different aspects of the data. Additionally, GPUs provide essential support for efficiently processing big data, enabling neural networks to scale and handle the computational demands of these complex datasets.

## Architecture

Depending on how are the layers, neurons, and connections structured in a neural network, they are grouped into different architectures Wei et al. (2020). The most basic one seen in examples everywhere, such as the one in figure 3, is a Multi-Layer Perceptron (MLP), or Feed Forward Neural Network (FNN). There is one input layer, some hidden layers, and one output layer, and that is the flow of information as well, without any loops or feedback connections. Recurrent Neural Networks (RNN), such as Long-Short Term Memory (LSTM) will add on the MLP and add recurrence and the possibility of storing information learnt in past nodes. For that reason, LSTM are strongly used in time-series forecasting (Kurt et al. 2008; Masood and Ahmad 2021; Rasp et al. 2020; Kaselimi et al. 2020). Among several other architectures, another relevant one is the Convolutional Neural Network (CNN), which is focused on visual data, for instance. These neural networks are developed in a way they are able to detect spatially related patterns in data through the use of several image processing filters with different kernel sizes. CNNs are hence strongly used for object detection, and image classification tasks, for example (Kattenborn et al. 2021; Cao et al. 2019; Du 2018).

When considering the goal to evaluate mainly spatial patterns in  $NO_2$  forecasts and having time and space considered in the modelling, a possible approach is the use of a Convolution Long Short Term Memory, or CNN-LSTM / ConvLSTM. The idea is to have both convolutions and a RNN in the same model, in a way that spatial structures can be assimilated through the convolution layers at the same time that as the temporal variable through recurrence (LSTM) (Liu, Mi, and Li 2018.; Shi et al. 2015; Wen et al. 2019).

The works of (Robin et al. 2022; Shi et al. 2015; Liu, Mi, and Li 2018; Samal et al. 2021; Beyer et al. 2023) demonstrate several applications of ConvLSTM and each of them present slight differences in the composition of the ConvLSTM architecture. How it is truly composed



in the end is usually a result of several tests given the specific purpose of the neural network. A general purpose is yet to be known and the testing approach is also expected to happen in the current work. Adaptation to the dataset, scale, temporal and spatial resolution are all characteristics to pay great attention when developing an architecture for a scoped network.

## Current Work

Once stated the context of Spatiotemporal Deep Learning and of pollution time series forecasting, it is then relevant to state what is the scope of this work and the scientific Dominion it is tied to. As mentioned, many researches focus on different approaches for pollution time series forecasting and only a few of them completely consider both space and time in modelling and even less consider the relevancy of weather when predicting such a correlated variable.

Those restricted articles then do not further analyse the spatial patterns in forecasts made by the models used. Do they follow the same spatio structure as the data before? Do they present heteroscedasticity throughout their surface? Is there a trend in future forecast in terms of space and time? All really relevant topics when working with spatial data, especially considering that the decision making concerning pollution may be present a strong spatial constraints marked by political boundaries.

## Objectives

Thus, this work is aimed at first investigating the feasibility of using SENTINEL 5P data to train a ConvLSTM neural network to forecast  $NO_2$  in both time and space, and, consequently, analyse whether this architecture suits this purpose when adding weather related covariables to it. A comparison is *ergo* required in terms of accuracy, but mostly considering spatial patterns in the errors.

Moreover, it is of extreme importance to evaluate the accuracy decay through time and, hence, verify, how far can predictions go into the future. The same logic is applied to space, verifying whether certain areas tend to be less accurate then others (heteroscedasticity).

A list of specific objectives can be seen below summarising the main targets present in this research.

- Use of a ConvLSTM to forecast  $NO_2$  with the aid of weather covariables
- Comparison of a ConvLSTM trained with weather covariables and another without in terms of accuracy and spatiotemporal heteroscedasticity.

## Research Questions

The study will be answering the following research questions:

- To what extent can one make use of a ConvLSTM neural network to forecast daily  $NO_2$  levels with SENTINEL 5P Imagery and weather related covariables?
- How does a ConvLSTM neural network using SENTINEL 5P and weather data performs compared to the same model without weather covariables, and how does the accuracy decay behaves when trying to predict further in time ?

## Methodology

The sequence of methods performed can be summarised by the data acquisition and wrangling of S5P data, including download, cloud cover removal and interpolation; followed by the ConvLSTM model development and fine tuning, ended by the validation and analysis of results. Consequently, weather data acquisition and cube merging with S5P data cube is done, which then also follows the steps of fine tuning the model and training with new data, to end with validation and analysis of the forecasts. Figure 4 below demonstrates these steps summarising the methods used to perform this analysis.

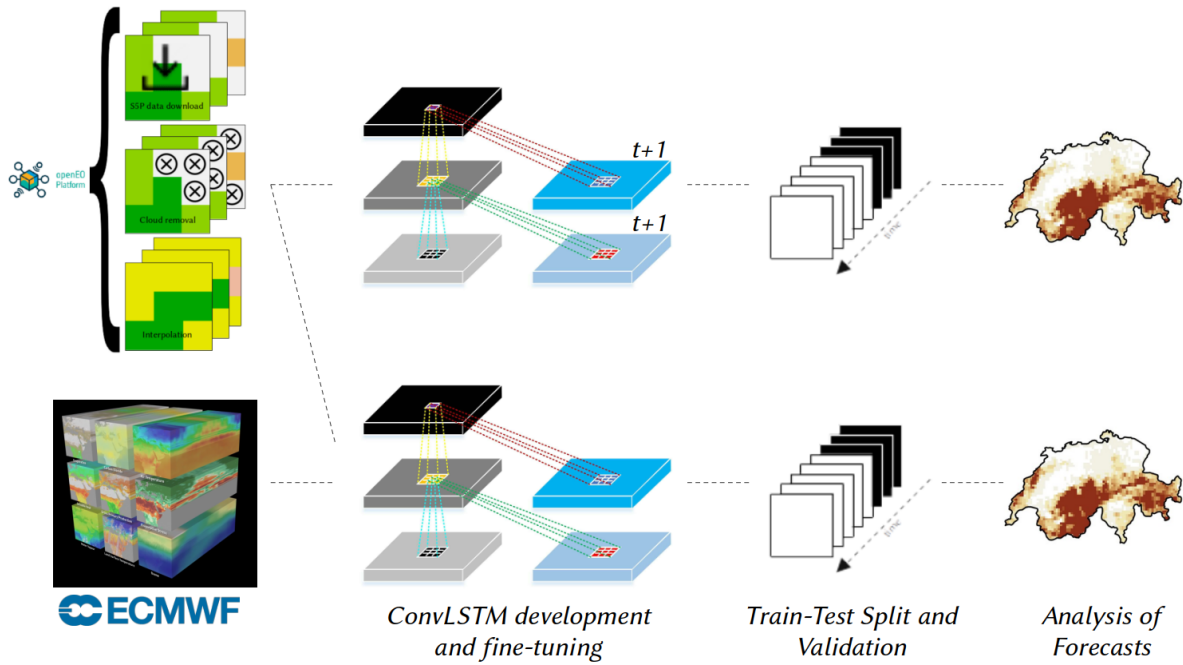


Figure 4: Summary of Methodology delineating the data acquisition and wrangling from both S5P and ERA5 data composing two exercises, one without weather data and the weather with it included. Partially adapted from Hu et al, 2019

## S5P Data Acquisition and Wrangling through openEO

The methodology started with SENTINEL 5P data acquisition and wrangling. This was done through the [openEO platform](#), which is a cloud service allowing big Earth data operations on data cubes, making use of many open Earth observation datasets. These operations can be done in different client programming languages, such as R, Python, Java or even a Web editor. Using openEO allows for straightforward acquisition through a simple bounding box and a timeline query and the following mask and interpolation operations. The mask refers to removing pixels that could be considered as clouds (disturbing the measured variable) and the interpolation is necessary to fill in the gaps created with the mask using the spatial and temporal neighbours of each empty pixel.

As a matter of fact, cloud remain to be a challenge in almost any remote sensing task. This operation in openEO allows for a trustworthy, but also straightforward approach. Nevertheless, the flag for the cloud cover removal remains open when dealing with SentinelHub data through openEO. This value roughly defines what should or not be considered a cloud in the image. As ESA recommends the use of 0.5 as a value, this has been used. As for the interpolation, a linear interpolation method has been applied.

The chosen area of study is Switzerland, and the vector of the country boundaries had been used as reference for the bounding boxes of the download from openEO. Data from the whole year of 2019 have been used for analysis. For reproducible purposes, inside the R programming language package built for the purpose of this research, the functions use to [download the data from openEO](#) and to wrangle them into a tensor for further learning are, respectively, `download_s5p_from_openeo` and `create_dl_from_cube`.

## ERA5 Data Acquisition through ECWMF

The download of ERA5 weather related variables has not been done in such an automatic way as for S5P one. ERA5 data is available through ECWMF and not yet integrated to openEO. The access to data has therefore been done through the [ECMWF webportal](#) and the data has been wrangled locally. The variables chosen for analysis were WInd Speed ( $m/s$ ), mean temperatures ( $K$ ), and Precipitation ( $m$ ). All data has been downloaded for the year of 2019 as well.

With the ERA5 data, the most important was to warp and resample it to fit the dimensions of the S5P  $NO_2$  data cube. This way, it was possible to create a cube similar of that of  $NO_2$  although with one more dimension related to weather. Still about that, the weather variables were summarised into one single, since the more dimensions there are, the higher the chances for over usage of RAM is. For this simplification, all of the three weather related variables were scaled from zero to a hundred and later the following logic was applied:

$$w = u_{10} * t_2 m - t_p$$

where:

- $w$  refers to the newly created weather related variable (the summarised value);
- $u_10$  refers to wind speed;
- $t_2m$  is the mean temperature;
- $t_p$  is the total precipitation.

Once more, for the purpose of reproduction of this step, [the functions in the R package that make reference to the wrangling of ERA5 data and their merge with the S5P data to build a tensor](#), respectively, are `read_ecwmfr_netcdf`, and `create_dl_with_weather`.

## Development of the Neural Network

To what refers to the Architecture built and its implementation, the whole model was developed in the R programming language, making use the `torch` (`torchlibrary?`) and helpers, such as the `luz` Falbel (2023a) and `torchvision` Falbel (2023b) packages. All of these tools are open-source and free to use ones, which allow for [full reproduction of the scripts developed for this work](#), given the necessary hardware demands.

These libraries and others such as `keras` and `tensorflow` are a few examples of tools created to allow for an easier and more straightforward implementation of deep learning models, including pre defined structures and architectures. In case of `torch`, the ConvLSTM is not implemented *per se*, although both the CNN and LSTM are already developed in a pretty high-level format. Hence, combining them is the most part of the programming challenge when referring to the model itself.

The ConvLSTM model is designed for analysing temporal data and consists of convolutional and LSTM layers. The architecture incorporates multiple layers, with each layer comprising a convolutional component followed by an LSTM component. A total of two ConvLSTM layers have been applied in this work, following other implementations for remote sensing purposes (Shi et al. 2015; Heydari et al. 2021; Wen et al. 2019; Samal et al. 2021). The convolutional layers extract spatial features from the input data, utilizing convolutional filters with a fixed kernel size of 3x3. These filters allow the model to detect local patterns and features in the data, capturing spatial dependencies and variations. The kernel size of 3x3 in the convolution layers ensures that the filters operate on a local receptive field, capturing fine-grained spatial patterns. This was also thought on the average pixel size, which is of around 4 km.

The LSTM layers in the ConvLSTM model capture temporal dependencies in the input data. LSTM units are equipped with input, forget, and output gates that regulate the information flow through recurrent connections. The hidden dimensions of the LSTM layers can be customized based on the requirements of the task, and here they were defined with 64 hidden dimensions.

Moreover, During training, the ConvLSTM model optimises its parameters, including the weights and biases of the convolutional and LSTM layers, to minimize a predefined loss function. This loss function is measured and graphed and it strongly indicated the capacity of the model to learn with each epoch, *i.e.*, a complete iteration of the model through the whole dataset.

The figure below, adapted from Hu et al. (2019), demonstrates shortly how the ConvLSTM is built. The convolution layers are passed with their kernel sizes on each pixel of the input data, and after words with the same pixel in the following time layer ( $t+1$ ). After that, the processes repeats itself and new weights are added to the nodes so loss can be diminished.

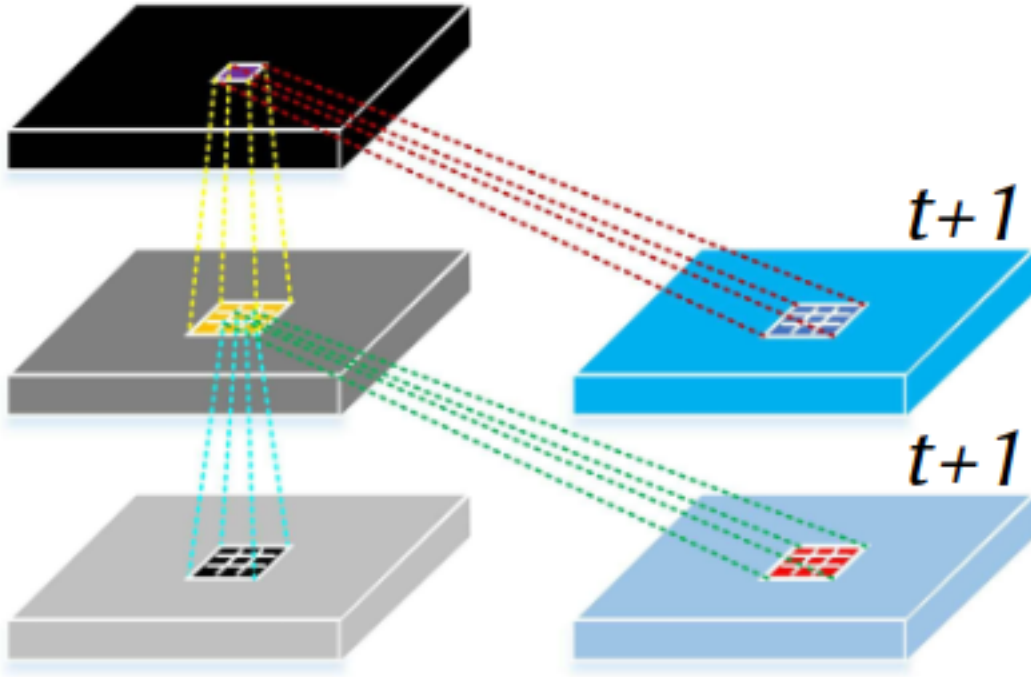


Figure 5: ConvLSTM structure demonstrating the convolution layers being passed to each pixel of the input data in current time and the next time reference. Two ConvLSTM layers are passed in total.

Finally, an important aspect of dealing with temporal models are windows. Windows, in this case, are temporal segmentations of the data, which each segment represents a contiguous sequence of consecutive time steps from the input data. Those are relevant in temporal models as it allows for capturing temporal dependencies and patterns in sequential data. The process of breaking the time series into windows enhances the granularity of the model, and thus support some sort of indirect seasonality and trend extraction from the data Chatfield (2000).

A problem with windows is that they are RAM demanding, and a reasonable number in those terms must be set. For this work, windows of size 12 and 4 have been set for the data without and with weather, respectively.

## Validation Procedure

Afterwards, in order to validate the results, a first thing to be done will be an analysis on the performance on the model, taking into account the time each model required to compile. This is important to understand what is the increase in demand of resources when using weather data.

As aforementioned, a deep look into the loss curve is also be conducted, so one can check for overfitting and the training capacity of the model. The data has been split into training and testing, so no interference between what is supposed to be known and unknown happens. This factor also tends to avoid overfitting, or at least help one detect it. Moreover, a train-test split of 80% / 20% has been taken into consideration, so the last months of data from the year of 2019 have been used for testing.

The main metric used for pixel-by-pixel error tracking is the Root Mean Squared Error (RMSE), which formula can be seen below:

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

where:

- $n$  is the number of samples in the dataset,
- $y_i$  represents the actual or observed value for the  $i$ -th sample,
- $\hat{y}_i$  represents the predicted value for the  $i$ -th sample,

The RMSE is a common metric used to measure the average difference between predicted and actual values in regression problems. It quantifies the root mean squared deviation between the predicted values and the ground truth, providing a measure of the model's performance. It is commonly used for analysing the performance of regression models, including deep learning ones. Hence, it is useful for comparison methods as well.

The RMSE will be evaluated as a mean for all forecasts, in an aggregated manner, but also per pixel and per unit of time and as a surface per unit of time and aggregated for all units of time. This separation shall allow for a deeper insights into spatial and temporal patterns in the errors, meaning that main targets, such as heteroscedasticity in spacetime can be thorough examined. Therefore, despite the RMSE, a qualitative observation of results is also done, checking whether errors present spatial dependence, for instance. In case clear spatial or temporal patterns are present in errors, a clear insight into the learning capacities of the ConvLSTM about these features shall be present.

## Hardware and Reproducibility

For computation comparison and reproducibility purposes, it is of extreme importance to detail what kind of hardware has been used for this work. This is especially relevant when considering the performance evaluations, for example. The main scripts have been developed in R programming language through a package that has been deployed in University of Münster High Performing Cluster (HPC) [PALMA](#). PALMA is a parallel Linux system with over 18000 processing cores, with access to multiple GPUs. The GPU used for this work was NVIDIA's TitanRTX2080 and overall 32GB RAM were requested. The processor of PALMA is Intel Xeon Gold 6140 18C @ 2.30GHz (Skylake).

## Results & Discussion

- Performance comparison
- RMSE (comparison to other works)
- Figure of errors
- RMSE aggregated.. (comparison- better because it makes sense)
- time series plot
- Spatial patterns plot

## Conclusion

- Agirre-Basurko, E., G. Ibarra-Berastegi, and I. Madariaga. 2006. "Regression and Multilayer Perceptron-Based Models to Forecast Hourly O<sub>3</sub> and NO<sub>2</sub> Levels in the Bilbao Area." *Environmental Modelling & Software* 21 (4): 430–46. <https://doi.org/10.1016/j.envsoft.2004.07.008>.
- Beyer, Martin, Rehaan Ahmad, Brian Yang, and Pablo Rodríguez-Bocca. 2023. "Deep Spatial-Temporal Graph Modeling for Efficient NDVI Forecasting." *Smart Agricultural Technology* 4 (August): 100172. <https://doi.org/10.1016/j.atech.2023.100172>.
- Cao, Changqing, Bo Wang, Wenrui Zhang, Xiaodong Zeng, Xu Yan, Zhejun Feng, Yutao Liu, and Zengyan Wu. 2019. "An Improved Faster r-CNN for Small Object Detection." *IEEE Access* 7: 106838–46. <https://doi.org/10.1109/access.2019.2932731>.
- Chatfield, Chris. 2000. *Time-Series Forecasting*. CRC press.
- Copernicus Climate Change Service. 2023. "Complete ERA5 Global Atmospheric Reanalysis." ECMWF. <https://doi.org/10.24381/CDS.143582CF>.
- Du, Juan. 2018. "Understanding of Object Detection Based on CNN Family and YOLO." *Journal of Physics: Conference Series* 1004 (April): 012029. <https://doi.org/10.1088/1742-6596/1004/1/012029>.

- Ebrahimi Ghadi, M., F. Qaderi, and E. Babanezhad. 2018. "Prediction of Mortality Resulted from NO<sub>2</sub> Concentration in Tehran by Air Q+ software and Artificial Neural Network." *International Journal of Environmental Science and Technology* 16 (3): 1351–68. <https://doi.org/10.1007/s13762-018-1818-4>.
- Falbel, Daniel. 2023a. "Luz: Higher Level 'API' for 'Torch'." <https://CRAN.R-project.org/package=luz>.
- . 2023b. "Torchvision: Models, Datasets and Transformations for Images." <https://CRAN.R-project.org/package=torchvision>.
- Freeman, Brian S., Graham Taylor, Bahram Gharabaghi, and Jesse Thé. 2018. "Forecasting Air Quality Time Series Using Deep Learning." *Journal of the Air & Waste Management Association* 68 (8): 866–86. <https://doi.org/10.1080/10962247.2018.1459956>.
- Grundström, M., C. Hak, D. Chen, M. Hallquist, and H. Pleijel. 2015. "Variation and Co-Variation of PM<sub>10</sub>, Particle Number Concentration, NO<sub>x</sub> and NO<sub>2</sub> in the Urban Air – Relationships with Wind Speed, Vertical Temperature Gradient and Weather Type." *Atmospheric Environment* 120 (November): 317–27. <https://doi.org/10.1016/j.atmosenv.2015.08.057>.
- Heydari, Azim, Meysam Majidi Nezhad, Davide Astiaso Garcia, Farshid Keynia, and Livio De Santoli. 2021. "Air Pollution Forecasting Application Based on Deep Learning Model and Optimization Algorithm." *Clean Technologies and Environmental Policy* 24 (2): 607–21. <https://doi.org/10.1007/s10098-021-02080-5>.
- Hu, Wen-Shuai, Heng-Chao Li, Lei Pan, Wei Li, Ran Tao, and Qian Du. 2019. "Feature Extraction and Classification Based on Spatial-Spectral ConvLstm Neural Network for Hyperspectral Images." *arXiv e-Prints*, arXiv–1905.
- Jo, Sungyang, Ye-Jee Kim, Kye Won Park, Yun Su Hwang, Seung Hyun Lee, Bum Joon Kim, and Sun Ju Chung. 2021. "Association of NO<sub>2</sub> and Other Air Pollution Exposures With the Risk of Parkinson Disease." *JAMA Neurology* 78 (7): 800. <https://doi.org/10.1001/jamaneurol.2021.1335>.
- Jutz, S., and M. P. Milagro-Pérez. 2020. "Copernicus: The European Earth Observation Programme." *Revista de Teledetección*, no. 56 (November). <https://doi.org/10.4995/raet.2020.14346>.
- Kaselimi, Maria, Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Demetris Delikaraoglou. 2020. "A Causal Long Short-Term Memory Sequence to Sequence Model for TEC Prediction Using GNSS Observations." *Remote Sensing* 12 (9): 1354. <https://doi.org/10.3390/rs12091354>.
- Kattenborn, Teja, Jens Leitloff, Felix Schiefer, and Stefan Hinz. 2021. "Review on Convolutional Neural Networks (CNN) in Vegetation Remote Sensing." *ISPRS Journal of Photogrammetry and Remote Sensing* 173 (March): 24–49. <https://doi.org/10.1016/j.isprsjprs.2020.12.010>.
- Kumar, Ujjwal, and V. K. Jain. 2009. "ARIMA Forecasting of Ambient Air Pollutants (O<sub>3</sub>, NO, NO<sub>2</sub> and CO)." *Stochastic Environmental Research and Risk Assessment* 24 (5): 751–60. <https://doi.org/10.1007/s00477-009-0361-8>.
- Kurt, Atakan, Betül Gulbagci, Ferhat Karaca, and Omar Alagha. 2008. "An Online Air Pollution Forecasting System Using Neural Networks." *Environment International* 34 (5):



- 592–98. <https://doi.org/10.1016/j.envint.2007.12.020>.
- Liu, Hui, Xiwei Mi, and Yanfei Li. 2018. “Smart Deep Learning Based Wind Speed Prediction Model Using Wavelet Packet Decomposition, Convolutional Neural Network and Convolutional Long Short Term Memory Network.” *Energy Conversion and Management* 166 (June): 120–31. <https://doi.org/10.1016/j.enconman.2018.04.021>.
- Lu, Meng, Marius Appel, and Edzer Pebesma. 2018. “Multidimensional Arrays for Analysing Geoscientific Data.” *ISPRS International Journal of Geo-Information* 7 (8): 313. <https://doi.org/10.3390/ijgi7080313>.
- Mahoney, Michael W., Mauro Maggioni, and Petros Drineas. 2006. “Tensor-CUR Decompositions for Tensor-Based Data.” *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August. <https://doi.org/10.1145/1150402.1150440>.
- Masood, Adil, and Kafeel Ahmad. 2021. “A Review on Emerging Artificial Intelligence (AI) Techniques for Air Pollution Forecasting: Fundamentals, Application and Performance.” *Journal of Cleaner Production* 322 (November): 129072. <https://doi.org/10.1016/j.jclepro.2021.129072>.
- Mohammadi, Mohammad Javad, Gholamreza Goudarzi, Sahar Geravandi, Ahmad Reza Yari, Bashir Ghalani, Saeed Shirali, Elahe Zallaghi, and Mehdi Esmaili. 2016. “Dispersion Modeling of Nitrogen Dioxide in Ambient Air of Ahvaz City.” *Health Scope* 5 (2). <https://doi.org/10.17795/jhealthscope-32540>.
- Panagakakis, Yannis, Jean Kossaifi, Grigorios G. Chrysos, James Oldfield, Mihalis A. Nicolaou, Anima Anandkumar, and Stefanos Zafeiriou. 2021. “Tensor Methods in Computer Vision and Deep Learning.” *Proceedings of the IEEE* 109 (5): 863–90. <https://doi.org/10.1109/jproc.2021.3074329>.
- Pawul, Małgorzata, and Małgorzata Śliwka. 2016. “APPLICATION OF ARTIFICIAL NEURAL NETWORKS FOR PREDICTION OF AIR POLLUTION LEVELS IN ENVIRONMENTAL MONITORING.” *Journal of Ecological Engineering* 17 (4): 190–96. <https://doi.org/10.12911/22998993/64828>.
- Pebesma, Edzer, and Roger Bivand. 2023. “Spatial Data Science,” March. <https://doi.org/10.1201/9780429459016>.
- Rasp, Stephan, Peter D. Dueben, Sebastian Scher, Jonathan A. Weyn, Soukayna Mouatadid, and Nils Thuerey. 2020. “WeatherBench: A Benchmark Data Set for Data-Driven Weather Forecasting.” *Journal of Advances in Modeling Earth Systems* 12 (11). <https://doi.org/10.1029/2020ms002203>.
- Rasp, Stephan, and Nils Thuerey. 2021. “Data-Driven Medium-Range Weather Prediction With a Resnet Pretrained on Climate Simulations: A New Model for WeatherBench.” *Journal of Advances in Modeling Earth Systems* 13 (2). <https://doi.org/10.1029/2020ms002405>.
- Robin, Claire, Christian Requena-Mesa, Vitus Benson, Lazaro Alonso, Jeran Poehls, Nuno Carvalhais, and Markus Reichstein. 2022. “Learning to Forecast Vegetation Greenness at Fine Resolution over Africa with ConvLSTMs.” <https://doi.org/10.48550/ARXIV.2210.13648>.
- Russo, Ana, and Amílcar O. Soares. 2013. “Hybrid Model for Urban Air Pollution Forecasting:

- A Stochastic Spatio-Temporal Approach.” *Mathematical Geosciences* 46 (1): 75–93. <https://doi.org/10.1007/s11004-013-9483-0>.
- Samal, K. Krishna Rani, Ankit Kumar Panda, Korra Sathya Babu, and Santos Kumar Das. 2021. “An Improved Pollution Forecasting Model with Meteorological Impact Using Multiple Imputation and Fine-Tuning Approach.” *Sustainable Cities and Society* 70 (July): 102923. <https://doi.org/10.1016/j.scs.2021.102923>.
- Shams, Seyedeh Reyhaneh, Ali Jahani, Saba Kalantary, Mazaher Moeinaddini, and Nematollah Khorasani. 2021. “Artificial Intelligence Accuracy Assessment in NO<sub>2</sub> Concentration Forecasting of Metropolises Air.” *Scientific Reports* 11 (1). <https://doi.org/10.1038/s41598-021-81455-6>.
- Shi, Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. 2015. “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting.” In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, 802–10. NIPS’15. Cambridge, MA, USA: MIT Press.
- Tsai, Yi-Ting, Yu-Ren Zeng, and Yue-Shan Chang. 2018. “Air Pollution Forecasting Using RNN with LSTM.” *2018 IEEE 16th Intl Conf on Dependable, Autonomous and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, August. <https://doi.org/10.1109/dasc/picom/datacom/cyberscitech.2018.00178>.
- Wei, Bing, Haibo He, Kuangrong Hao, Lei Gao, and Xue-song Tang. 2020. “Visual Interaction Networks: A Novel Bio-Inspired Computational Model for Image Classification.” *Neural Networks* 130 (October): 100–110. <https://doi.org/10.1016/j.neunet.2020.06.019>.
- Wen, Congcong, Shufu Liu, Xiaojing Yao, Ling Peng, Xiang Li, Yuan Hu, and Tianhe Chi. 2019. “A Novel Spatiotemporal Convolutional Long Short-Term Neural Network for Air Pollution Prediction.” *Science of The Total Environment* 654 (March): 1091–99. <https://doi.org/10.1016/j.scitotenv.2018.11.086>.
- Wood, E. C., S. C. Herndon, T. B. Onasch, J. H. Kroll, M. R. Canagaratna, C. E. Kolb, D. R. Worsnop, et al. 2009. “A Case Study of Ozone Production, Nitrogen Oxides, and the Radical Budget in Mexico City.” *Atmospheric Chemistry and Physics* 9 (7): 2499–2516. <https://doi.org/10.5194/acp-9-2499-2009>.
- Zhang, Zili, Jian Wang, and Wenju Lu. 2018. “Exposure to Nitrogen Dioxide and Chronic Obstructive Pulmonary Disease (COPD) in Adults: A Systematic Review and Meta-Analysis.” *Environmental Science and Pollution Research* 25 (15): 15133–45. <https://doi.org/10.1007/s11356-018-1629-7>.
- Zhao, Zhong-Qiu, Peng Zheng, Shou-Tao Xu, and Xindong Wu. 2019. “Object Detection with Deep Learning: A Review.” *IEEE Transactions on Neural Networks and Learning Systems* 30 (11): 3212–32. <https://doi.org/10.1109/tnnls.2018.2876865>.
- Zhu, Min, Jing Xia, Xiaoqing Jin, Molei Yan, Guolong Cai, Jing Yan, and Gangmin Ning. 2018. “Class Weights Random Forest Algorithm for Processing Class Imbalanced Medical Data.” *IEEE Access* 6: 4641–52. <https://doi.org/10.1109/access.2018.2789428>.