

Table of contents

1	Abstract	2
2	Introduction	3
2.1	Background	4
2.1.1	Weather and Pollution Forecasting	4
2.1.2	Time Series Forecasting	5
2.1.3	Spatial Data Cubes and Sentinel 5P	5
2.1.4	Spacetime Deep Learning	8
2.1.5	Architecture	9
2.2	Current Work	10
2.2.1	Objectives	10
2.2.2	Research Questions	11
3	Methodology	11
3.1	S5P Data Acquisition and Wrangling through openEO	13
3.2	ERA5 Data Acquisition through ECWMF	13
3.3	Development of the Neural Network	14
3.4	Validation Procedure	16
3.5	Hardware and Reproducibility	17
4	Results & Discussion	17
4.1	Learning Rate	18
4.2	Loss decay & Performance	20
4.3	Validation	20
4.4	Spatio-temporal heteroscedasticity	24
4.5	Future Work	25
5	Conclusion	27
	References	28

1 Abstract

Nitrogen dioxide (NO_2) is a greenhouse gas strongly related to respiratory diseases, often used as a reference for pollution indicators. Monitoring the presence of this gas in the atmosphere is therefore of high importance, and the same can be said of its forecasting. For that purpose, computing the spatial and temporal dependencies all together can be a resourceful way, given the nature of the phenomenon. Hence, a Convolutional Long Short-Term Neural Network (ConvLSTM), a deep learning architecture focused on combining convolutional and long short-term memory layers for combining space and time dimensions was developed to forecast Sentinel 5P NO_2 data. In addition to space and time, this network is highly capable of handling high-dimensional data, which favours the possibility of including different covariates in the model. Hence, the model could include weather-related variables that are highly correlated with NO_2 formation, such as wind speed, direction, temperature, humidity, precipitation, atmospheric pressure, and solar radiation. A deep learning model was trained and evaluated using not only time-series data but also the series of satellite imagery from Sentinel 5P for daily NO_2 forecasting. Second, a similar architecture was trained using weather time-series and forecasts from ERA5 (ECWMF), which drove the comparison to the first model in terms of accuracy. The study aimed to answer 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 perform compared to the same model without weather covariables, and how does the accuracy decay behave when trying to predict further in time? 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, ending with 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. Results for the model without weather presented an average RMSE of 0.033, whilst the weather model presented an RMSE of 0.113. In general, the model with the weather was more homoscedastic in both space and time, although it tended to over-smooth the resulting surfaces, ignoring detailed information.

2 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. 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 covering the entire earth, which makes it a substantial possibility for a worldwide 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, despite 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, such 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 accuracy in daily forecasts.

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

2.1 Background

2.1.1 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 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 temperature, the higher the chance for NO_2 presence (Wood et al. (2009)). Other factors, 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 $PM_{2.5}$, another pollutant indicator, making use of weather-related variables as supporters, being wind speed, temperature, wind direction, rainfall, and humidity. In their work, this relationship is tested in those models and it is left 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.

2.1.2 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, within a data-driven approach, the logic is to use past information to assume the future. Any new information will be given based on the 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 is 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. However, 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, perhaps 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 a decision has to be made regarding pollution, this decision has boundaries associated with 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 that is forgotten in the model or in the results, or weather that is not being considered as a covariable. A gap in the literature is therefore present.

2.1.3 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 in 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 is constantly imaging Earth's and other planets surface for full coverage data. These include 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 derived 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, which supports a highly accurate capture of pollutants such as NO_2 , O_3 , CH_2O , SO_2 , CH_4 and CO .



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

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 to 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 (2023b) 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 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.

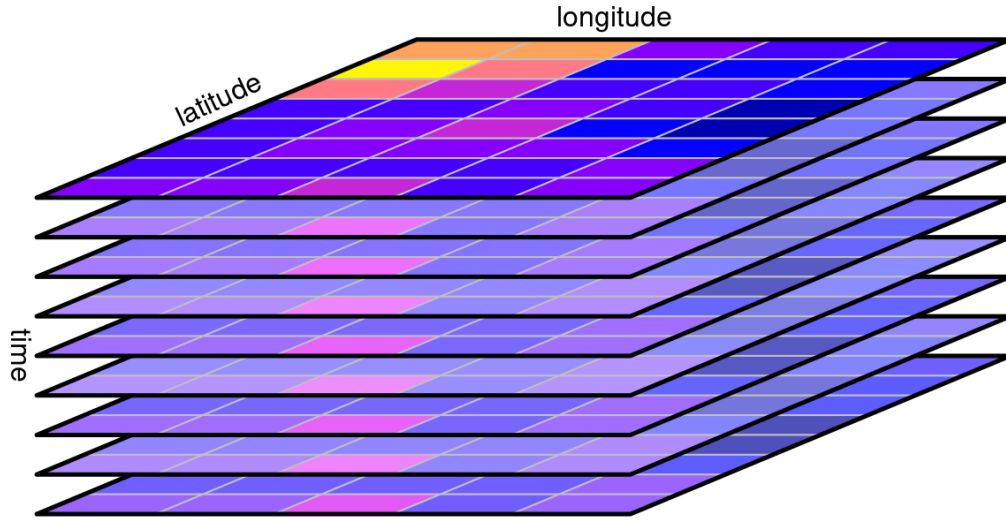


Figure 2: Representation of a Spatial Data Cube. Source. Pebesma and Bivand, 2023

The structure of the data cube is especially interesting for 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.

2.1.4 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 to approximate desired input-output mappings. Figure 3 demonstrates the neurons and weights that are programmatically iterated to reach optimal results.

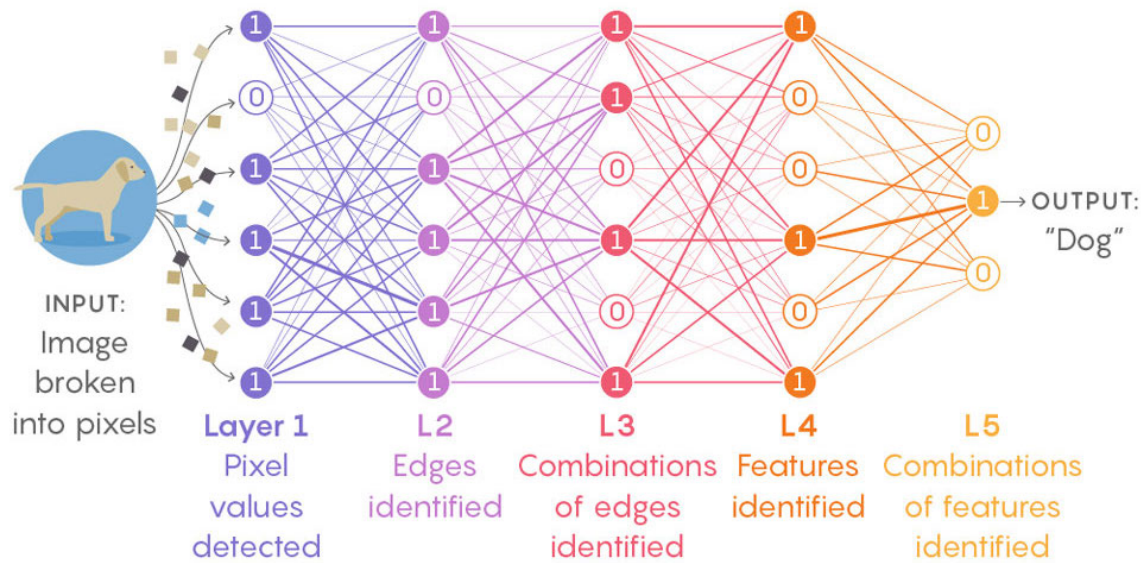


Figure 3: Neural Network model, represented with the objective of image classification. Source: Wolchover, Natalie (2019)

The exercise present in Figure 3 is that of an image classification. An input image of a dog is given as an input array and it be processed by the different layers, which nodes will be giving iterative weights until the array is correctly assigned the desired label. Of course, this requires several inputs to be reachable, hence being a considerably 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 Panagakis 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.

2.1.5 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, LSTMs 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 an RNN in the same model, in a way that spatial structures can be assimilated through the convolution layers at the same time 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 to when developing an architecture for a scoped network.

2.2 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 research works 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 fewer 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 spatial structure as the data before? Do they present heteroscedasticity throughout their surface? Is there a trend in future forecasts in terms of space and time? All relevant topics when working with spatial data, especially considering that the decision-making concerning pollution may be present as strong spatial constraints marked by political boundaries.

2.2.1 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 than 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.

2.2.2 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 ?

3 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, ending with 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.

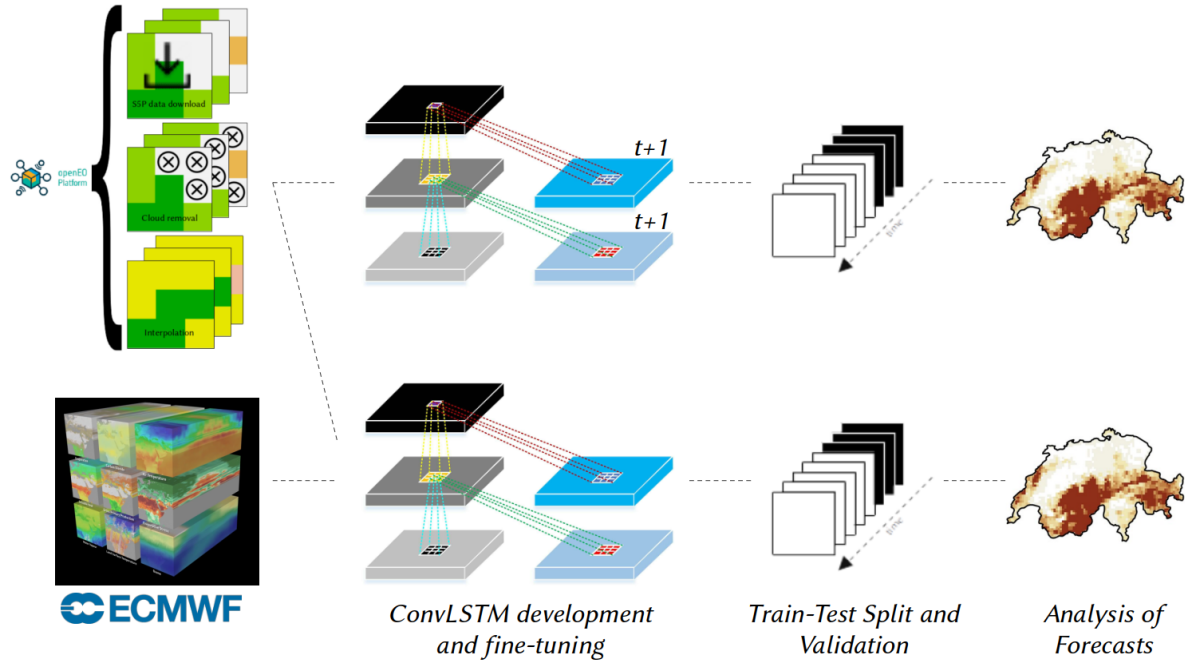


Figure 4: Summary of Methodology delineating the data acquisition and wrangling from both S5P and ERA5 data composing two exercises, one without weather data and the weather with it included. Partially adapted from Hu et al, 2019

3.1 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, clouds remain 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 as 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.

It is relevant to mention that after the wrangling was processed through the *openeo* package Lahn (2022), a local cleaning has also been made. The S5P data as found in Sentinel Hub present a considerable level of what is called *Pepper and Salt*. This effect refers to the lack of smoothness in Earth observation data, with outliers breaking the general patterns in images. For that reason, a blurring algorithm has also been applied to the data in order to smooth out these outliers and create a cleaner and probably more realistic surface to forecast.

The chosen area of study is Switzerland, and the vector of the country boundaries was used as a 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 this research, the functions used 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`.

3.2 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 web portal](#) 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 to 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 three weather-related variables were scaled from zero to a hundred and later the following logic was applied:

$$w = u_{10} * t_2m - t_p$$

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 reproduction of this step, [the functions in the R package that refer to the wrangling of ERA5 data and their merge with the S5P data to build a tensor](#), respectively, were `read_ecwmfr_netcdf`, and `create_dl_with_weather`.

3.3 Development of the Neural Network

What refers to the Architecture built and its implementation, the whole model was developed in the R programming language, making use of 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 the 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 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 indicates 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 afterwards with the same pixel in the following time layer ($t+1$). After that, the process repeats itself and new weights are added to the nodes so loss can be diminished.

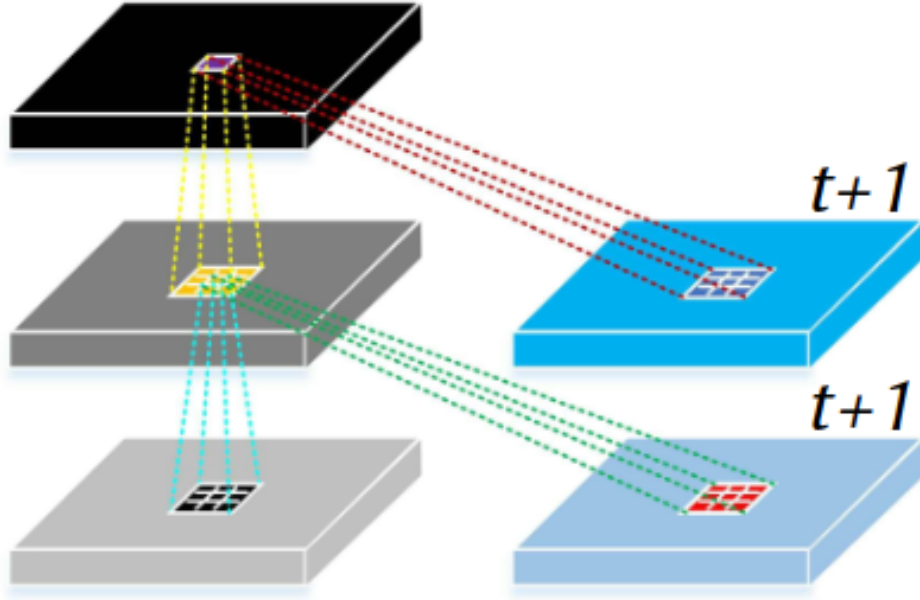


Figure 5: ConvLSTM structure demonstrating the convolution layers being passed to each pixel of the input data in current time and the next time reference. Two ConvLSTM layers are passed in total.

After that, an important aspect of dealing with temporal models is windows. Windows, in this

case, are temporal segmentations of the data, in which each segment represents a contiguous sequence of consecutive time steps from the input data. Those are relevant in temporal models as they allow 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 supports 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, a window of size 30 days has been set for the data without and with weather.

Finally, when developing the hyperparameter tuning, *i.e.*, setting the parameters of the model that will bring the most optimal outcome, one of those that is exceptionally calibrated through a more specific manner other than testing is the learning rate. The learning rate defines the size of the steps, at which the optimisation algorithm updates the model's parameters during training Smith et al. (2017) . The choice of the learning rate is usually done by comparing it to the loss when shortly training the model for only one or two epochs. This procedure was also done and the final result was used for the actual model training.

3.4 Validation Procedure

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

As aforementioned, a deep look into the loss curve is also 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 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 a deeper insight into spatial and temporal patterns in the errors, meaning that main targets, such as heteroscedasticity in spacetime can be thoroughly 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.

3.5 Hardware and Reproducibility

For computation comparison and reproducibility purposes, it is of 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 a Linux PC with 40 GB RAM and 4 cores with an Intel i7 10th generation processor. A GPU was planned to be used, but at the end, CPU was capable of handling computations, as comparison was still feasible.

4 Results & Discussion

All code developed for this work has been made inserted into an R package. This allows for full reproducibility and also relative easiness in applying the same model to different data in the future. The package is available in [GitHub](#) freely, just as the data.

First of all, data from S5P with NO_2 layer has been obtained through the `openeo` package in R, where cloud cover removal and interpolation were also made. A further cleaning process has been made through a blurring algorithm from the `spatstat` package Baddeley, Rubak, and Turner (2015). This was especially relevant to avoid a “Salt-and-Pepper” effect and having over-detailed imagery for model training. An example of this de-blurring can be seen in the figure below.

One can observe that the overall spatial pattern of the NO_2 plume is kept, especially to what concerns local maximums and minimums, and the outliers are completely wiped out of the image. More than that, areas where even the interpolation could not fill due to cloud cover were finally smoothed in a similar way. In general, the objective of smoothing out the data was obtained and that proved relevant for a reasonable loss decay once the model was trained.

The development of the ConvLSTM Network was followed by the adaptation of past work and several testing drills. The main challenge in setting up the neural network has been the data alignment and structuring. Having it in the correct format as input for the model and finally,

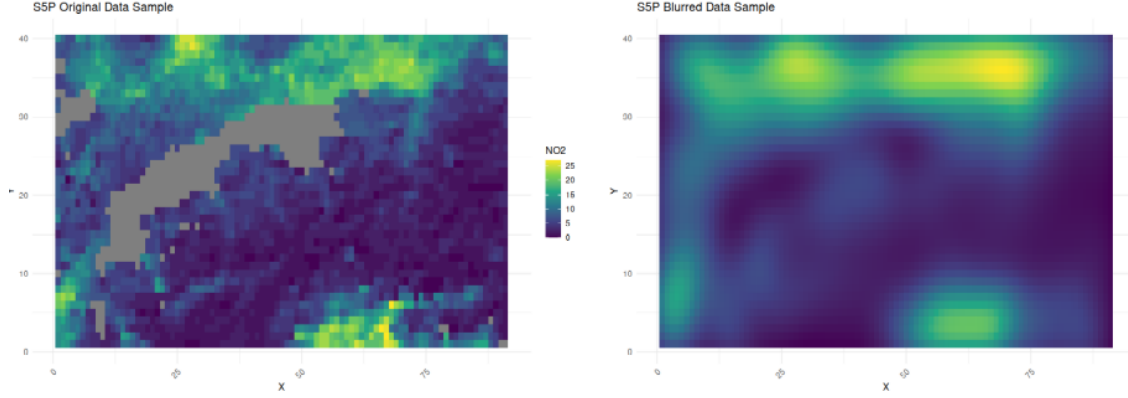


Figure 6: Demonstration of the blur effect applied to the S5P before the model training process. As one can see, the main spatial pattern is maintained and missing data is also roughly eliminated.

the output can come as a simple challenge, although this was not the case. When handling images and having them converted to high dimensional matrices, many issues showed up in the path. Even for a rather shallow neural network, the data wrangling and testing of the training given different hyperparameters was the most time-demanding task, therefore not to be underestimated. Still on the training, some steps are considerably important and dealt with in the following.

4.1 Learning Rate

Once data was in its expected format, the learning rate curve was computed so one could identify an optimal value for it. In the following, one can verify that an approximate optimal learning rate of around 0.01 was found, as this refers to the area in the x-axis where the loss is the most stable and the lowest at the same time. Hence the value of 0.01 was used for model training and prediction.

Nevertheless, once the model was tested with different parameters and finally demonstrated reasonable results, it was time to download the weather data from ECMWF. As mentioned once before, the download of the weather data was done on the website of ECMWF. The images were translated into matrices with the main aid of the `stars` package Pebesma and Bivand (2023a), where they were also *warped* to fit the dimensions of the S5P data. They were converted into a single variable to ease computation and a higher dimensional matrix was set to be inputted into the ConvLSTM neural network. As the model was already coherently developed and tested with the S5P data, almost no adaptation was required in this step regarding the neural network.

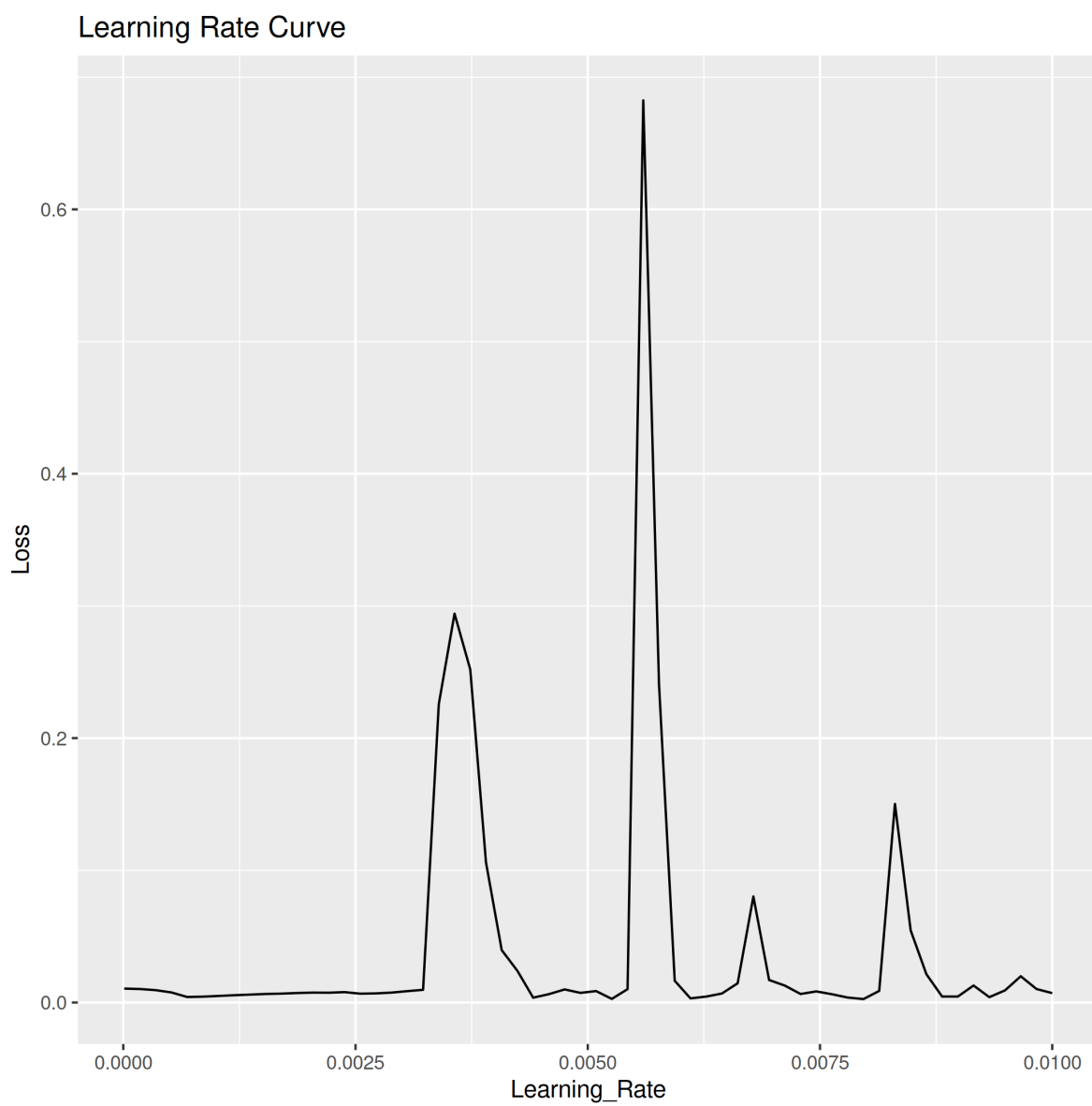


Figure 7: Learning Rate Curve which allowed to identify the optimal learning rate value of 0.01

4.2 Loss decay & Performance

To have a deeper look into the learning process of the ConvLSTM, the loss curve of the model trained without weather data and with weather data can be seen below.

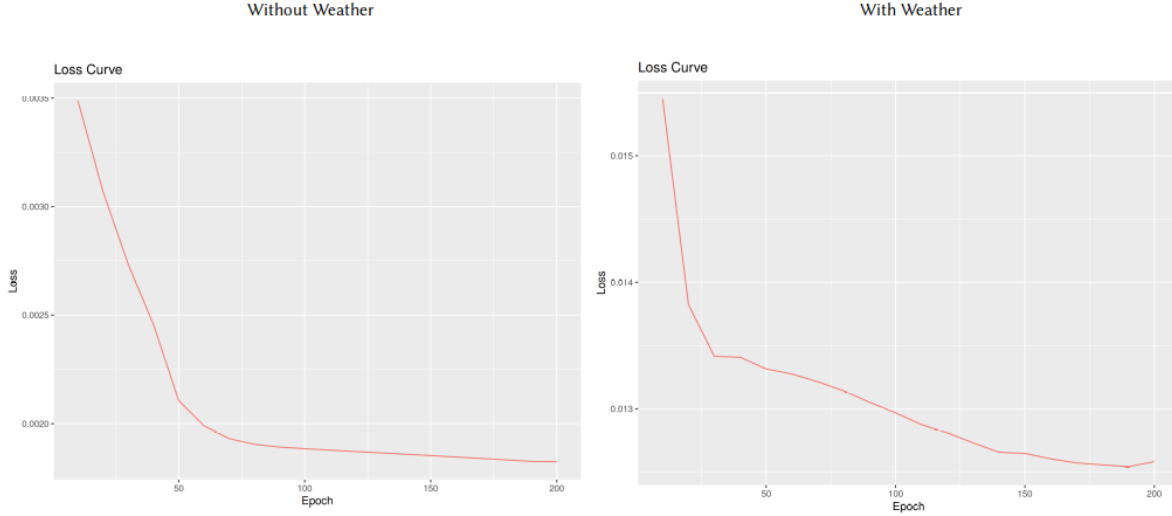


Figure 8: Loss curve of model without and with weather data

As one can see, the loss decays smoothly until reaching an asymptote, just as expected from a well-trained deep learning model. Other than that, the smoothness is considerably different between the two models. The model without weather data presents a considerably smoother loss curve than the model trained with weather data. This could represent a stronger difficulty in training when having a higher dimensional array, but also the possibility that an even more specific tuning was required. In the end, the curve was still decaying and it also reached a low asymptote. However, it is the validation metric and the other methods that should actually evaluate how helpful the weather was in forecasting pollution.

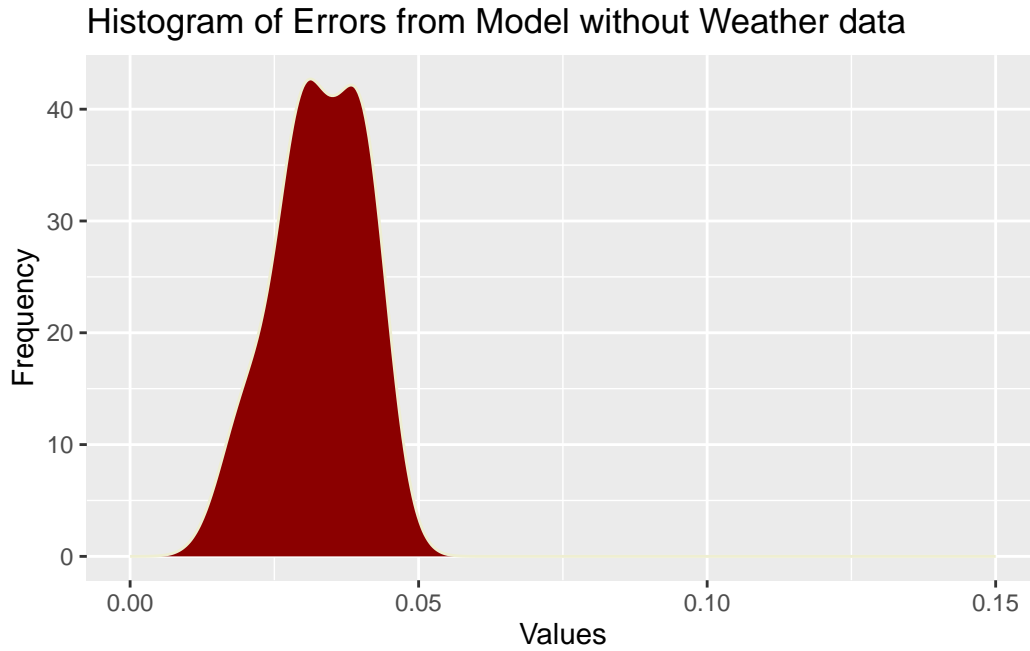
Moreover, 200 epochs were used and a slight difference in training time was noticed. Whilst training the model without weather data took approximately 48 min, the model with weather data required no more than one hour to complete. It is important to mention that this difference could be due to background processes interacting with the CPU and the RAM. Nevertheless, one can roughly say the training is rather more time-demanding when using higher-dimensional data, but not to the point it would turn it into a relevant drawback.

4.3 Validation

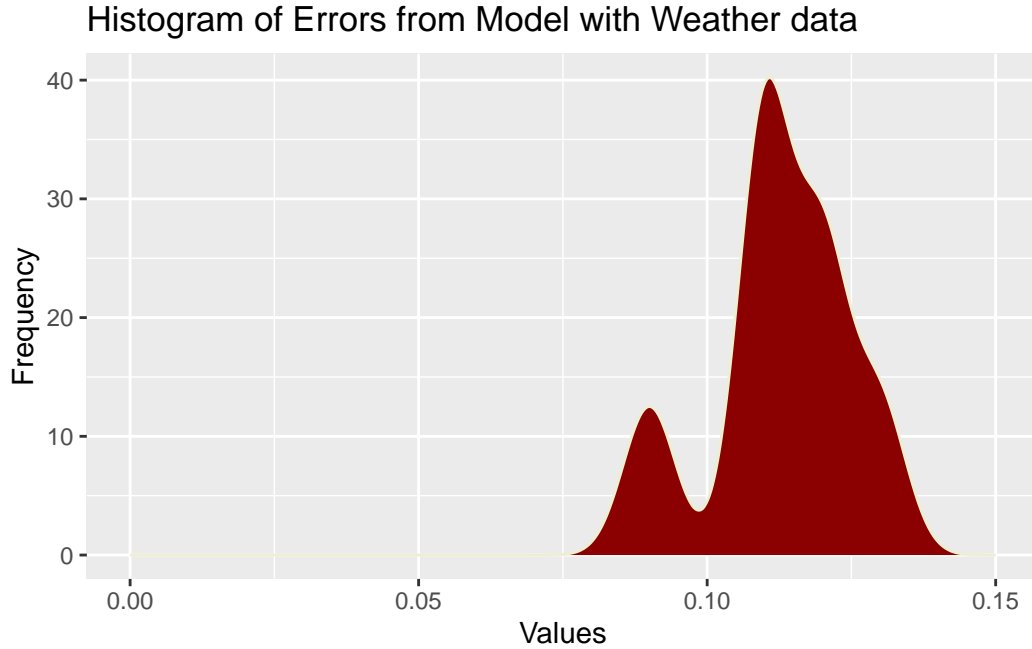
The validation process was followed given the attention to the RMSE metric, but also the main point of spatial and temporal heteroscedasticity. When ignoring the weather data, an RMSE

of 0.033 was found, *i.e.*, the accuracy was around 97% among the total of all timestamps forecasted (seven days), given that the data was scaled from zero to one. The obtained results are relatively comparable to those presented by Wen et al. (2019) and Shi et al. (2015) given into account the different units and also even targets. The loss obtained was 0.0018 in the end, which will then be compared to the loss of the model trained with weather data.

The histogram below demonstrates the RMSEs obtained for the seven predictions made, which shall give a clue about how spread the errors were. One can observe that the errors roughly show a normal distribution and they were indeed concentrated around the mean RMSE. The maximum error found was 0.04 (4%) and the minimum of 0.02 (2%).



Meanwhile, when looking at the predictions from the model trained with weather data, RMSE and loss were considerably higher. The obtained loss in this case was of 0.0126, whilst the RMSE was of 0.113 (11%). Similarly, the histogram of errors coming from the predictions of the model trained with weather data shows a concentration of errors further in the x-axis. This time, the histogram falls apart from a normal distribution and demonstrates two modes. Hence, this model has presented to be less accurate, but also less certain.



These results have exposed a different response than the one expected, as weather data was supposed to support the predictions of NO_2 , as mentioned by Grundström et al. (2015) and Samal et al. (2021). The outcome of this comparison could lead one to believe that the weather is not a main driver of NO_2 , although this correlation may not represent causation. It is possible the feature engineering developed was not adequate for this data, as well as the fact that the architecture may not be set the most appropriately to work with covariables. Given that, in the figure below, one can also compare the weather-featured data and the NO_2 data found in the training data set.

Following the figure, there is some relationship between the variables, although it is not explicitly evident. Nevertheless, a neural network should be able to compute such a non-linear relationship, which does not support the hypothesis that weather does not drive pollution. What must be considered is that the neural network one is dealing with is a motion capture neural network, so it is important to verify how NO_2 compares to each other to what refers to time. The following figure demonstrates the predicted timeslots and the weather surface for each of them.

This time, when looking at the temporal pattern, one can see that indeed the space is quite well respected in terms of the relationships between weather and NO_2 , although temporally, the weather-featured engineered data may not help explain pollution as well as expected. Not only has it an inferior spatial resolution, but the temporal symbiosis between the phenomena may not be well captured in the relatively small study window, but also it may significantly decrease the resolution of the pollution forecast: an important output of this work.

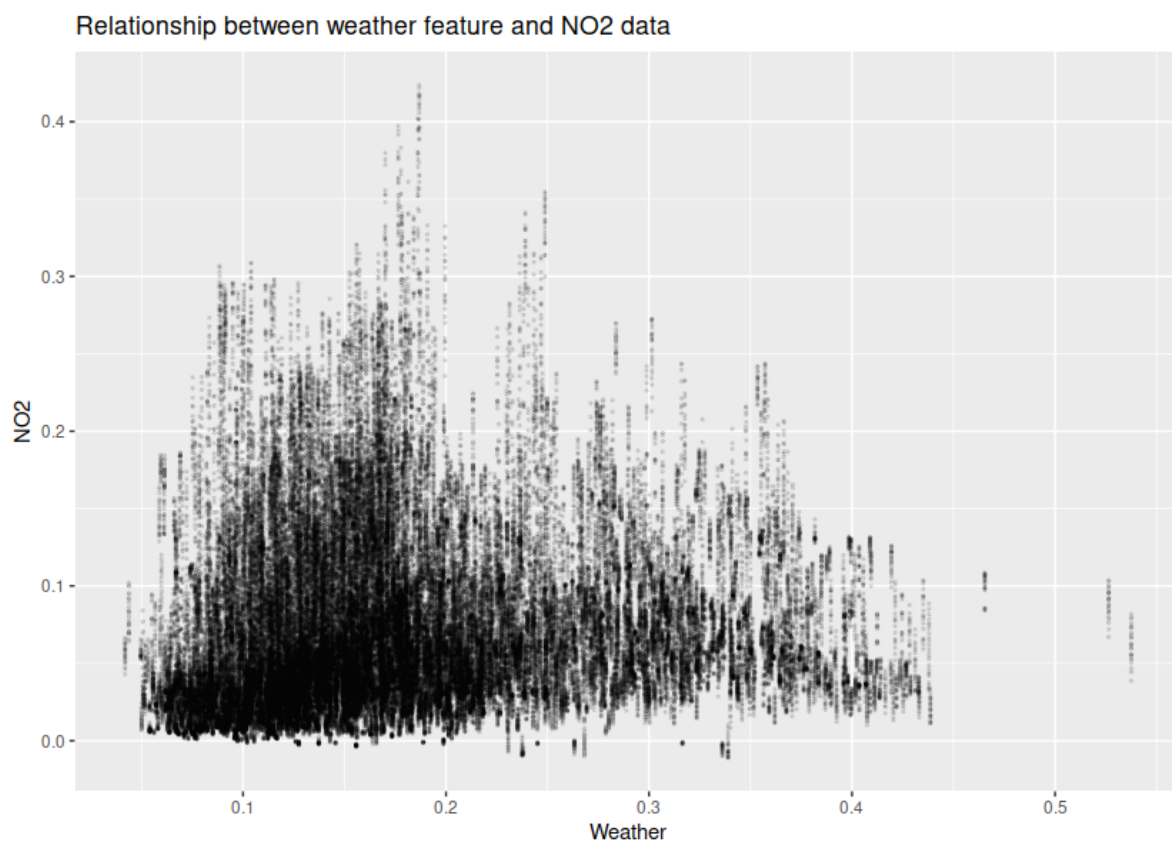


Figure 9: Relationship between weather featured data and NO2

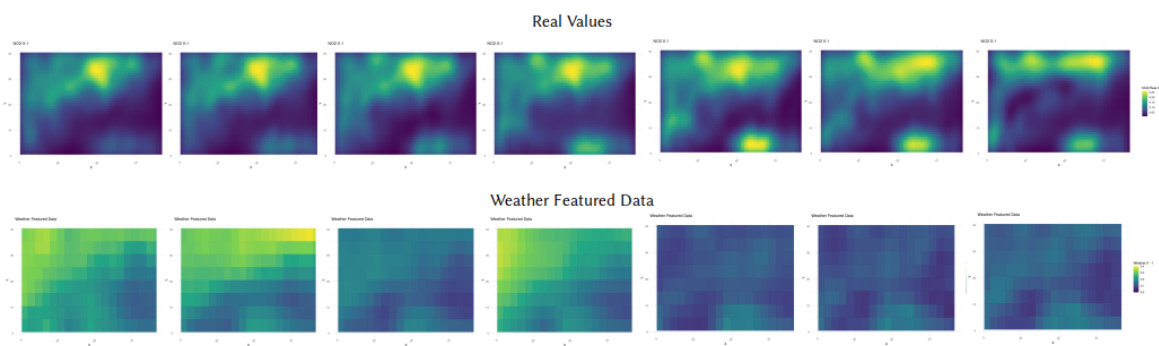


Figure 10: Temporal relationship between weather-related data and NO2 pollution

4.4 Spatio-temporal heteroscedasticity

The final and perhaps most central part of this work was then to discuss the heteroscedasticity in space and in time, *i.e.*, how the predictions and their errors behave in both space and time. In order to bring the spotlight to this topic, the next figure shall support one in the analysis of both dimensional patterns.

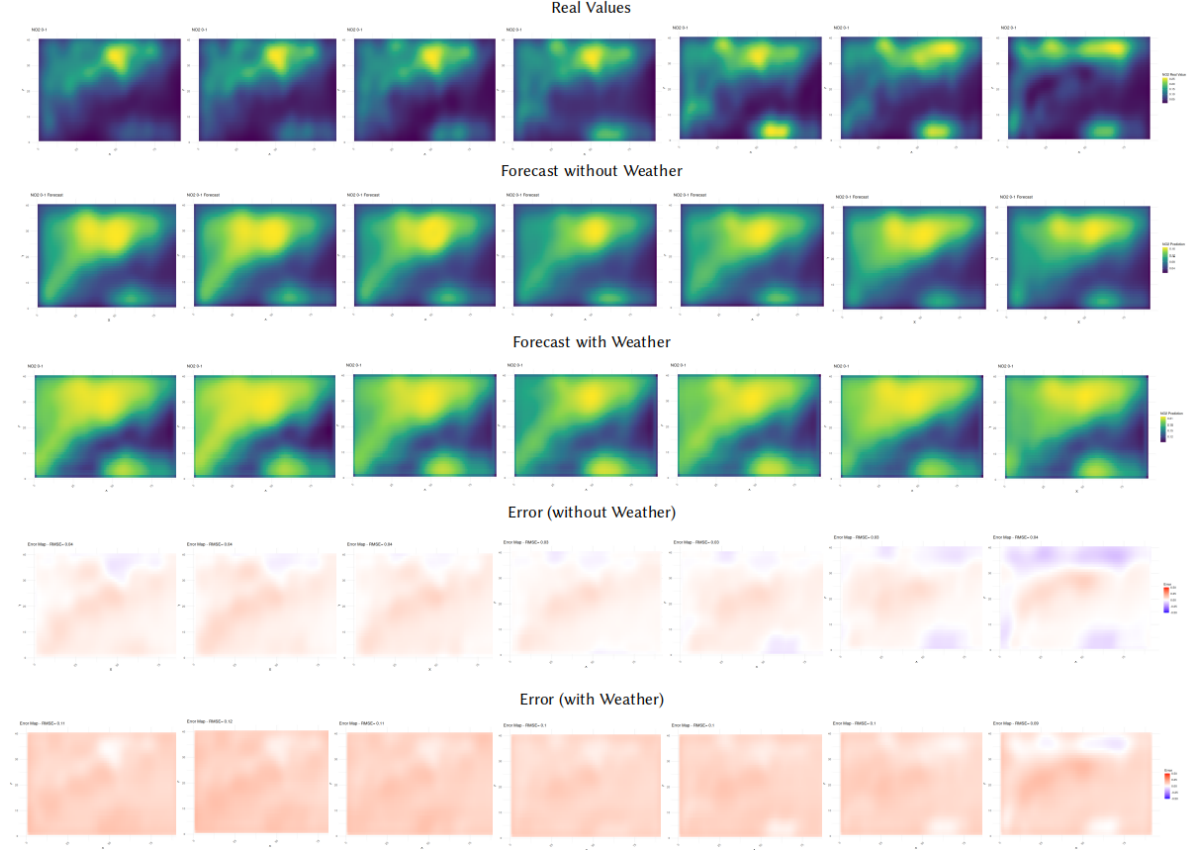


Figure 11: one in the analysis of both dimensional patterns. Comparison between real values and predictions made with models using and not using weather data, together with their errors

Taking a closer look at the spatial behaviour of the forecasts, one of the first characteristics that call one's attention is the identification of local maximum and minimum values. All models at all times tend to generally respect the overall characteristic of the original data, especially at the most extreme points. Nevertheless, the differences start to get more evident when one observes the degree of smoothness. The model trained without weather data is already considerably smoother than the original data (which was blurred *per se*), which is a phenomenon that only gets stronger when looking at the results of the model trained with weather data. The weather data had a lower spatial resolution and this can have affected the

results negatively as they tend to over-simplify the predicted surface. Therefore, the weather data we not able to support the inferencing of detailed changes in space or time, which presents the most difficult challenge, also when observing the model trained without weather data. Even so, when not using weather, more details are captured. In the 5th and 6th time-steps present in the figure above, one can see that a high point gets even stronger in the south of the image. The model without weather data is slightly capable of detecting this change, whilst the model with weather is not.

Still looking at space, one specific area calls one attention, the dark blue in the centre of the images. This area had originally missing data, which were only covered with the blurring effect once described. This area, spatially speaking, is the one concentrating the most errors, in both models, especially in the last period. Although, one can notice that the model with the weather has a slighter difference comparing this area to the rest, being spatially more homoscedastic. Hence, one can infer that making use of the weather covariable considerably supports filling in missing data, demonstrating a still present relationship between weather and atmospheric pollution.

Furthermore, when looking at the time, the model that does not make use of the weather tends to have a stronger decay in accuracy with time than the model that makes use of the weather covariable. This heteroscedasticity in the temporal dimension is not seen to the same extent in the outcomes of the weather-based model. The errors there tend to be indeed higher there, although temporally more stable. This may indicate that the weather gives further support in understanding and interpreting changes in time. The weather covariable may be, in other words, an aid in the forecast task *per se*.

Finally, still in the topic of spatio-temporal heteroscedasticity, making use of a weather-related covariable diminished the irregular errors in both dimensions. However, when talking about errors *per se*, there is a clear over-smoothing of the resulting predictions. So, when coming back to the question of whether using a weather-related covariables is useful or not, the answer will mostly depend on one's need. To forecast NO_2 data coming from Sentinel 5P in both space and time, weather supports lower heteroscedasticity, but it tends to increase overall errors and the used supporting covariable has a lower spatial resolution.

4.5 Future Work

Further analysing the present work, it is important to state what suggestions could be made in order to drive further research in the topic of NO_2 spatio-temporal forecasting. First of all, one of the greatest challenges one is presented to when dealing with time series is to handle future covariables. In this case, in a real-life scenario, in the future one can only have weather forecasts instead of weather historical data, which may have different results. For that task, one has to use recent data, have a way to structure past and future weather data and then validate it only only the time has passed.

Furthermore, as seen in the results, the topic of filling in missing data is always a topic in any data science scenario, but especially in remote sensing, as clouds are omnipresent. In this case, linear interpolation and blurring were both used, which also helps to have a more complex logic underneath, which tends to guarantee the model is not learning to do interpolation only. Nevertheless, further tests in this direction can be interesting, as the topic of cloud cover is still relevant in the field.

Finally, something that was only briefly explored during tests was the use of each weather variable separately, instead of a feature-engineered one. Tests have shown few differences, but further exploring the multiple possibilities, especially with a more alike spatial resolution when compared to the predictor variable could be a topic of further exploration.

5 Conclusion

In the course of this work, a convolutional-LSTM was developed to forecast NO_2 atmospheric pollution in Switzerland with the aid of weather-related covariables. The predictions of this model were fully compared when using weather and when not using weather in terms of accuracy, but mainly spatial and temporal heteroscedasticity. Hence spatial and temporal patterns or predictions and errors were thoroughly analysed.

Shortly, the development of the Conv-LSTM was guided through a series of tests and adaptations that finally led to a smoothly falling loss curve with the aid of the learning rate tuning. Data wrangling was the most time-demanding task as fitting the data in the tensor format is not a straightforward process.

In summary, one could observe that using weather as a covariable supported more homoscedastic predicting surfaces in both space and time. Errors were generally more homogeneous when using weather, but the fact that the spatial resolution of weather data was smaller than that of the S5P visibly contributed to having an overall lower accuracy when using weather data. Moreover, very likely due to the difference in spatial resolution again, the level of detail when not using weather as a covariable was considerably higher than when using weather data, which was already less detailed than the validation data. Moreover, when looking more into the temporal dimension, the model without weather data presented a stronger accuracy decay through time than the one with weather data. In that sense, weather data presented to be a useful resource to help understand temporal changes in the pollution phenomenon, although it occurred to have over-smoothed the forecast surface.

For further work, it is recommended to look into three main topics: the use of real weather forecast data instead of weather historical data in the future, requiring a different validation procedure; analyse to what extent cloud removal can still affect the neural network learning process and to which level is it not learning to interpolate it simply; and the use of each weather variable individually instead of a feature engineered variable.

References

- Agirre-Basurko, E., G. Ibarra-Berastegi, and I. Madariaga. 2006. "Regression and Multilayer Perceptron-Based Models to Forecast Hourly O₃ and NO₂ Levels in the Bilbao Area." *Environmental Modelling & Software* 21 (4): 430–46. <https://doi.org/10.1016/j.envsoft.2004.07.008>.
- Baddeley, Adrian, Ege Rubak, and Rolf Turner. 2015. "Spatial Point Patterns: Methodology and Applications with {r}." <https://www.routledge.com/Spatial-Point-Patterns-Methodology-and-Applications-with-R/Baddeley-Rubak-Turner/p/book/9781482210200/>.
- 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 NO₂ 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 PM₁₀, Particle Number Concentration, NO_x and NO₂ 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 NO₂ 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 (O₃, NO, NO₂ 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>.
- Lahn, Florian. 2022. “Openeo: Client Interface for ‘openEO’ Servers.” <https://CRAN.R-project.org/package=openeo>.
- 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/10.1016/j.hsc.2016.02.001>.

[//doi.org/10.17795/jhealthscope-32540](https://doi.org/10.17795/jhealthscope-32540).

- Panagakis, Yannis, Jean Kossaifi, Grigorios G. Chrysos, James Oldfield, Mihalīs 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. 2023a. “{Spatial Data Science: With Applications in r},” 352. <https://doi.org/10.1201/9780429459016>.
- . 2023b. “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 NO₂ 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.
- Smith, Samuel L., Pieter-Jan Kindermans, Chris Ying, and Quoc V. Le. 2017. “Don’t Decay the Learning Rate, Increase the Batch Size.” <https://doi.org/10.48550/ARXIV.1711.00489>.

- 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, Autonomic 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>.