Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft Institute of Applied Mathematics

Using Artificial Intelligence for Aerosol Data Assimilation

(Dutch title: Het gebruik van Artificiële Intelligentie voor Aerosol Data Assimilatie)

Thesis submitted to the
Delft Institute of Applied Mathematics
in partial fulfillment of the requirements

for the degree

MASTER OF SCIENCE
in
APPLIED MATHEMATICS

by

GUUS VAN HEMERT

Delft, Nederland
August 2022

MSc report APPLIED MATHEMATICS

“Using Artificial Intelligence for Aerosol Data Assimilation”
(Dutch title: “Het gebruik van Artificiële Intelligentie voor Aerosol Data Assimilatie “

GUUS VAN HEMERT

Delft University of Technology

Supervisors

Dr.ir. H. X. Lin
Dr. O. Hasekamp
Dr. A. Tsikerdekis

Committee

Dr. M. B. van Hoven

August, 2022 Delft

Abstract

To study the aerosols in the atmosphere is an important aspect for getting a better understanding of climate change. Therefore, it is important to get accurate observations of aerosols in the atmosphere as well as accurate emission fluxes of aerosol species. Satellite instruments such as SPEXone are able to measure aerosol properties with a high accuracy. Unfortunately, the instrument has a low daily global coverage. To obtain full daily global coverage, methods such as data assimilation are used. However, these methods have a high computational cost. This report investigates the use of neural networks to obtain global daily coverage of aerosol properties and emission fluxes with a lower computational cost. Two networks are trained. One to get global coverages of the aerosol properties Aerosol Optical Depth at 550nm (AOD), Single Scattering Albedo at 550nm (SSA) and Ångström Exponent between 550nm and 865nm(AE). The other network is trained for global emission fields of the species dimethylsulfide (DMS), sulfur dioxide (SO_2), black carbon (BC), organic carbon (OC), sea salt (SS) and dust (DU). The results from these trained networks are compared to the results of a control experiment, which represents our prior knowledge on the aerosol fields and emissions, although not the truth. It is found that the network for aerosol properties has a significant decrease in errors compared to the control experiment. For both AOD and AE, the network has a large improvement, and for SSA the improvement is slightly smaller, likely due to a lower performance of the control experiment compared to AOD and AE. The network for the emissions also has a noticeable improvement over the control experiment for all species except DMS, where there is only a small improvement due to the already accurate DMS value for the control experiment. It is also found that the network for emissions overfits due to too little variation in training and testing data.

Contents

Chapter 1

Introduction

1.1 Problem Statement

Satellite measurements are a crucial part of analyzing and understanding the effect of aerosols on the climate. The spatial coverage of satellite measurements is determined by factors. The orbital properties of the satellite and cloud coverage are all factors on the coverage of the measurements. Since aerosols can only be measured over cloud free regions and on average 67% of the Earth is covered by clouds, no satellite exists that is able to provide full daily global coverage, even for instruments with a very large field of view [1]. Moreover, it is impossible to directly retrieve aerosol emissions with a satellite instrument, since an instrument is only able to measure certain properties of the aerosols. To obtain accurate information of aerosol properties for the entire globe, data assimilation methods are used. These methods combine the available satellite measurements with atmospheric global climate models that describe emissions, transport an formation/loss process of aerosols in the atmosphere. One drawback of these methods is the computational cost, which is high due to the complexity of these atmospheric global climate models. Therefore, it is important to find an alternative method with good accuracy and a lower computational time. In this report, the use of artificial intelligence, in particular deep learning, will be implemented to investigate the capabilities of a neural network and if it can be used as a substitute for data assimilation. Neural networks are known for their low computational cost, since the network needs to be trained only once, after which it is able to compute predictions very fast. The problem will be formalized in Section 1.3, after a brief introduction on aerosols.

1.2 Introduction to Aerosols

Before we can formalize the problem statement, we first give a short introduction on aerosols. This will be partially based on an article from Myhre et al. [2]. Atmospheric aerosols are suspensions of liquid, solid, or mixed particles with highly variable chemical composition and size distribution. We make the distinction between two types of aerosols. Primary aerosols, which are directly emitted into the atmosphere, and secondary aerosols, which are produced in the atmosphere from precursor gases. Primary aerosols consist of both organic and inorganic components. The inorganic components can originate from sea spray, mineral dust and volcanic activities, while the organic components come from vegetation and micro-organisms and anthropogenic sources. Examples of anthropogenic sources are combustion processes and biomass burning. These combustion processes are sources of carbonaceous aerosols, which include organic carbon (OC) and solid black carbon (BC). BC is the largest anthropogenic light-absorbing contributor in aerosols. Secondary aerosols consist of compounds with the main components sulphate, nitrate and some species of OC. The largest part of the precursor gases are emitted from the combustion of fossil fuels (anthropgenic sources), but fires also play an important part.

Aerosols are known to have an impact on the climate. All aerosols scatter incoming solar radiation and a few species also absorb solar radiation. BC is an important factor in the absorption, however mineral dust and some OC components are also able to absorb the radiation. Aerosols which mainly scatter sunlight have a cooling effect on the Earth, while aerosols which have a strong absorption have a warming effect. The atmosphere contains a mixture of scattering and absorbing aerosols and their net effect on the radiation balance depends on surface and cloud characteristics. For example, scattering aerosols above a dark surface and absorbing aerosols above a bright surface are most efficient. Conversely, scattering (absorbing) aerosols above a bright (dark) surface are less efficient since the surface reflects (absorbs) the sunlight anyway. Absorbing aerosols are particularly efficient when they are above clouds, which are a main contributor for reflecting solar radiation. Overall, scattering by aerosols dominates over the absorption by aerosols. A bright surface has a so-called high albedo, while a dark surface has a low albedo.

Aerosols play a vital part in cloud formation, since a portion of them can serve as cloud condensation nuclei (CCN) and ice nuclei (IN). An increase in the amount of aerosol can also increase the CCN number concentration and lead to more but smaller cloud droplets. This in turn increases the albedo of the cloud, which results in a higher reflection and cooling effect. Aerosols can also modify the cloud properties, without directly acting as CCN and IN. For example, this can be done by heating the air surrounding them while reducing the amount of solar radiation reaching the ground. This stabilizes the atmosphere which reduces the potential for cloud formation. Alternatively, aerosols can increase the atmospheric temperature leading to a reduced relative humidity, inhibits cloud formation and further enhances the evaporation of the existing clouds.

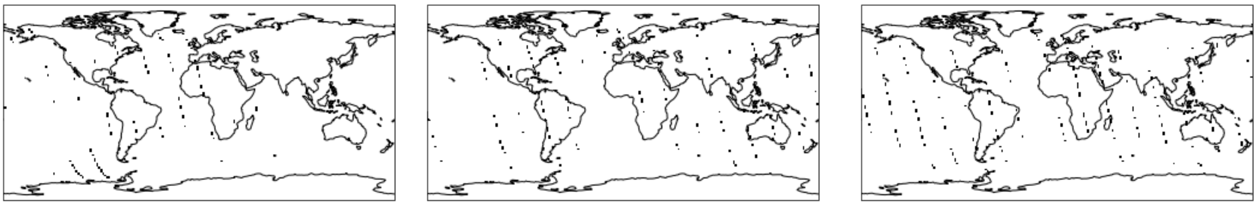
To quantify and compare the potential impact on the climate of various aerosol effects, radiative forcing (RF) is often used. RF is the change in the radiation balance of the Earth as a result of perturbations of anthropogenic or natural origin. With the help of RF and recorded temperature changes of the Earth, the climate sensitivity can be determined. Therefore it is important to study the aerosols and their effect in the atmosphere, such that RF can be determined which in return can be used to study climate changes.

There are various properties of aerosols which can be measured. In this report we will focus on three of those properties. The first way to quantify aerosols is by the Aerosol Optical Depth (AOD). AOD is a very common measurement for aerosols and it is related to the amount of light that is scattered or absorbed by aerosols in a columns through the atmosphere. The AOD is dependent on wavelength and the most common wavelength reported by satellite data products is 550 nm. The second quantity is the Single Scattering Albedo (SSA), which is related to the reflection and absorption properties of the aerosols. More precisely, it is the fraction of light that is scattered compared to the total extinction optical depth. The total extinction in this instance is the combination of scattering and absorption. This means that an SSA of 1 indicates that an aerosol only scatters light, while a value of 0 means it mostly absorbs the light. As with AOD, SSA is also dependent on wavelength. Lastly, the Ångström Exponent (AE) is a measure of the change of the AOD over various wavelengths of light. AE is related to the aerosol particle size. Roughly speaking, a value less than 1 corresponds to a higher presence of coarse particles like dust or sea spray, while values greater than 1 correspond to a dominance of fine particles like smoke.

Although measurements of these optical properties give an indication of aerosol type, they do not give direct information on aerosol emission sources. Since it is important to relate aerosol properties to emission sources, for example to reduce carbon emissions at a potential big anthropogenic source, we would also want to be able to derive these emission from the measurements.

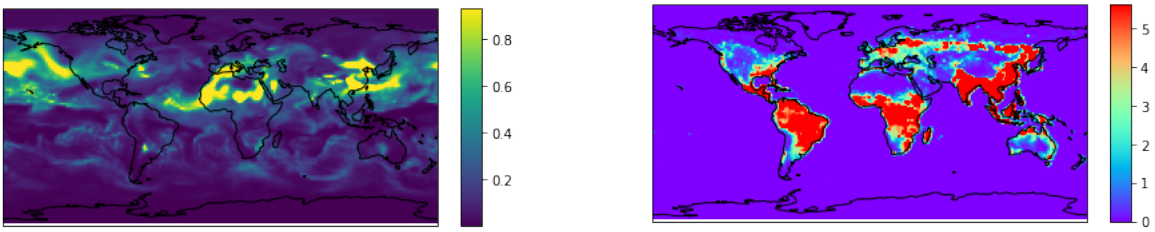
1.3 Formalization of the Problem

With the short overview of aerosols and some of their properties, we can now formalize the problem. As stated in Section 1.1, this report will focus on the use of a neural network to obtain accurate aerosol properties from satellite measurements for the entire globe. The simulated satellite measurements are from the SPEXone instrument, which will be launched with the NASA PACE mission scheduled to launch in 2024. The SPEXone instrument gives very accurate measurements, however it has limited coverage in a day [3] (see Figure 1.1). The paper from Tsikerdekis et al. is taken as the starting point for this project [4]. Tsikerdekis et al. used an ensemble Kalmann smoother to estimate aerosol emissions from SPEXone measurements in combination with the ECHAM-HAM atmospheric model. Since the SPEXone instrument has not been launched, no real observations are available. To circumvent this issue, they performed Observing System Simulation Experiments (OSSEs). In OSSEs, a model simulation, in this case from a simulation of ECHAM-HAM with perturbed emissions, is considered to be the truth, also called the nature run. From this simulation, synthetic measurements are sampled based on the on the spatiotemporal coverage of the satellite instrument. Afterwards, two other experiments are performed, a control experiment and a data assimilation experiment. The control experiment is again a model simulation, but with different results than the nature run, by either using a different model or by using the same model with different emissions and/or physics options. The data assimilation experiment uses both this control experiment and the simulated measurements. These two experiments can then be compared to the considered truth to investigate the performance of data assimilation with the observations. Tsikerdekis et al. assimilated AOD, SSA and AE. Both AOD and SSA were at a wavelength of 550 nm and AE was for the wavelengths between 550 nm and 865 nm. With these assimilated results, they estimated the aerosol emission fluxes for desert dust (DU), sea salt (SS), OC, BC and SO_4 along with two precursor gases for SO_4, namely SO_2 and dimethylsulfide (DMS).



(a) Poorly observed day. (b) Typical day. (c) Well observed day.

Figure 1.1: The coverage of the SPEXone instrument for a poorly observed day (5th percentile), a typical day (median) and a well observed day (95th percentile).



(a) Global AOD field of one day. (b) Global field of OC emissions for one day.

Figure 1.2: Global fields of both AOD and OC emissions of a specific day.

The setup of our experiments will be very similar to the setup in [4]. We will also make use of OSSEs in the sense that we assume one model simulation to be the truth and sample the synthetic measurements from this model simulation. Likewise, we have a control experiment that should represent out prior knowledge on aerosol fields and/or emissions. This thesis will focus on:

1. To construct a neural network which will predict global information on AOD, SSA and AE from SPEXone measurements and the control experiment results.
2. To construct a neural network which will predict global emission fluxes of DMS, SO_2, BC, OC, SS and DU from AOD, SSA and AE fields from the trained network from problem 1.

The performance of the trained models will be compared to the true global fields of aerosol properties and emissions and the improvement compared to the control run will be assessed.

1.4 Organization of the Report

The remaining of the report will be structured in the following way. In Chapter 2 a short introduction on neural networks will be provided. Next we will discuss other works related to our problem in Chapter 3. Then datasets used for the two problems and the used neural network architecture are discussed in Chapter 4 and 5 respectively. In Chapter 6 and 7 the results of problem 1 and 2 respectively are discussed. Finally, in Chapter 8 the report is concluded.

Chapter 2

Neural Network Prerequisites

In this chapter, some theory about neural networks, with the emphasis on Convolutional Neural Networks, will be given. Section 2.1 on neural networks in general will be inspired on an article from Kriegeskorte and Golan [5], while the information on the convolutional networks in Section 2.2 is taken from Albawi et ak. [6].

2.1 Neural Networks

Neural networks, or artificial neural networks, are inspired by the workings of biological neurons. Neural networks consist of an input layer, where it takes multiple inputs, one or more hidden layers and an output layer. Each layer is composed of nodes, or neurons, which are connected to the nodes of the subsequent layer. Each node has an assigned weight and threshold. The weight of a node determines the importance of that node in predicting the output, while the threshold manages the activation of the node. If the output of a node is above the threshold for that node, its output is sent to the next layer of the network.

Once an input layer is created, weights are assigned to each input in that layer. All inputs are multiplied by their respective weight and then summed. Next, the output is passed through an activation function, which gives the output of the layer. If this output exceeds the threshold, the output is passed to the subsequent layer. For a node with a threshold c receiving m inputs we have the following

$$f(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} w_i x_i \geq c, \\ 0, & \text{if } \sum_{i=1}^{n} w_i x_i < c, \end{cases} \tag{2.1}$$

where w_i is the weight of input x_i and $f(x)$ is the activation function of the node. However, this activation function has a derivative of zero, making it harder to optimize. Therefore, other activation functions are often chosen. Two commonly used activation functions are the sigmoid activation function $f(x) = \frac{1}{1+e^{-x}}$ and the ReLU activation function $f(x) = \max(0, x)$. The last layer in the network, the output layer, the outputs of the previous hidden layer are combined, again using an activation function, to obtain a function which maps the input space into the output space. This is done by minimizing a cost or loss function, like for example the mean squared error. The more hidden layers the network has and the bigger the size of these layers, the better it is able to learn complex relations.

In general, there are two types of learning, namely supervised learning and unsupervised learning. If the cost function minimized during training is a function of the output of the network and the real output, it is called supervised learning. If the cost function is not dependent on the real output, it is called unsupervised learning.

The training process of a network, in this case supervised, works as follows. First, the weights are initialized. The weights are then slightly adjusted in each step to get closer to the desired output. For each input, the activations are propagated through the network, where the activation state is computed for each unit in the network, including the output. At the output layer, the output from the network is then compared with the desired output and the cost is computed, using the cost function. Then the partial derivative of the cost with respect to the activation is computed to check how much the cost can be reduced with a slight change in the activation. This partial derivative is then propagated back through the network to the activations of the previous layer. As the activations are dependent on the weights, since $f(x) = f(\boldsymbol{wx})$, the derivative of the cost with respect to the weights can be computed using the chain rule. With the help of this derivative, the weights can be adjusted in the direction that reduces the cost. This also shows why it is preferred for the activation function to have a non-zero derivative, as otherwise the derivative would vanish, causing troubles for the weight adjustments.

2.2 Convolutional Neural Networks

Convolutional Neural Networks, or CNN's in short, are a special type of artificial neural networks that excel in pattern recognition. Its main advantage over classic artificial neural networks is that it greatly reduces the amount of parameters, allowing the model to solve larger problems. With this property, it is mainly used for image problems. The network typically takes an image with shape $H \times W \times D$ as input. This input image is then traversed by a filter, or kernel, with a shape of $f \times f \times D$, performing matrix multiplications with the part of the image the filter highlights at the time. The outcome of this multiplication is summed and then projected onto the feature map, which is passed to the next layer. This can be seen in Figure 2.1, where we have an image of shape $8 \times 8 \times 1$ and a kernel of size $3 \times 3 \times 1$. In this case, the kernel is at the top left where it performs a matrix multiplication with the highlighted part of the image. The outcome of this action, which is a vector, is then summed to get a scalar value, which is then projected onto the feature map. The filter starts at the top left corner of the image and moves over the image to the right with steps equal to the so-called stride value until it reaches the boundary of the image. After it reaches the boundary of the image it starts at the left of the image again and moves down with another stride value, which is typically the same stride value. It then moves to the right again and repeats this process until the entire image is traversed.

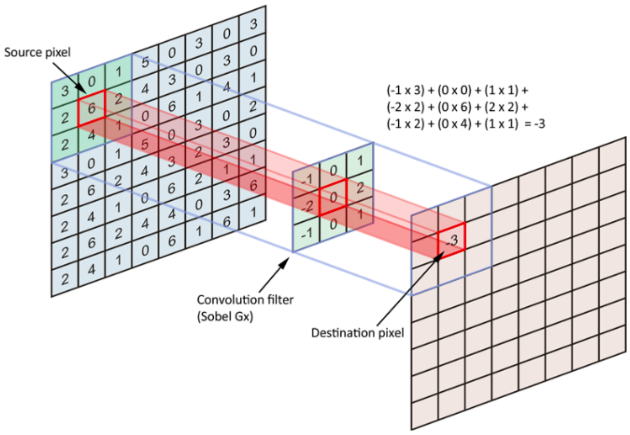


Figure 2.1: The operation of a $3 \times 3 \times 1$ kernel on a $8 \times 8 \times 1$ input image [7].

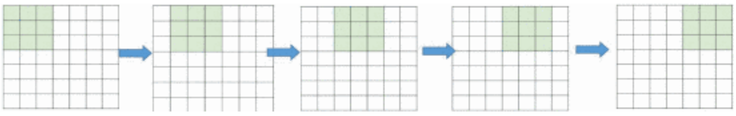
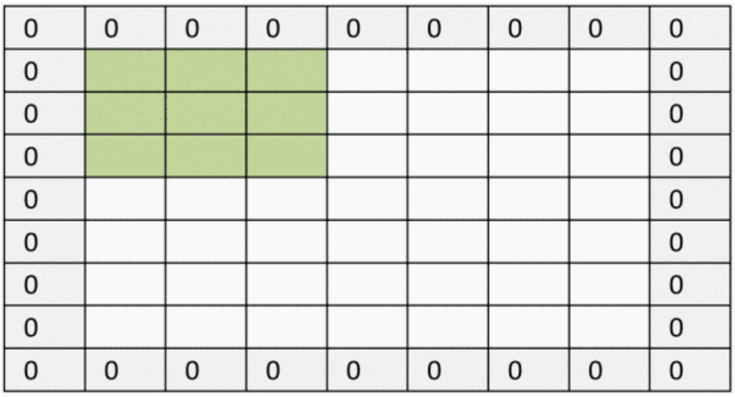


Figure 2.2: A kernel which traverses the image with a stride of 1. After the last position of the filter, it will start at the left again, 1 below the original position [6].

Figure 2.2 shows how the filter traverses the image if it has a stride of 1. However, considering that we have a 7×7 image with a 3×3 filter and a stride of 1, the output image will have a reduced size of 5×5. This effect can also be seen in Figure 2.1, where the output of the first filter is projected on position (1,1) instead of position (0,0). Generally, if we have an image of size $N \times N$ and a filter with size $F \times F$ with stride S, the output image will be of shape $O \times O$, where $O = 1 + \frac{N-F}{S}$. Therefore, to allow the output image to retain its original shape, zero-padding is used on the input image. With zero-padding, the input of size $N \times N$ will be increased to $(N+2) \times (N+2)$ and the boundaries will be set to zero. This will not only make sure the images do not decrease in size, it also helps to prevent information loss at the boundaries.



0	0	0	0	0	0	0	0	0
0	🟩	🟩	🟩					0
0	🟩	🟩	🟩					0
0	🟩	🟩	🟩					0
0								0
0								0
0								0
0								0
0	0	0	0	0	0	0	0	0

Figure 2.3: Zero-padding on an image of size 7 × 7 [6].

Lastly, we would like to consider pooling layers. Pooling layers down-sample the input image, which reduces the complexity of the image for subsequent layers. The advantage of this is that the model will become easier to train as it reduces the number of parameters that need to be trained. There are two types of pooling, namely max-pooling, which is the most common, and average-pooling. In both pooling types, the input image is partitioned into smaller rectangles. If max-pooling is applied, the maximum of each sub-rectangle is returned, while for average-pooling the average is returned. The most common size for the sub-regions is 2 × 2 with a stride of 2. This means that both the width and the height of the image get halved in the pooling-layer, which can also be seen in Figure 2.4 , where max-pooling is used.

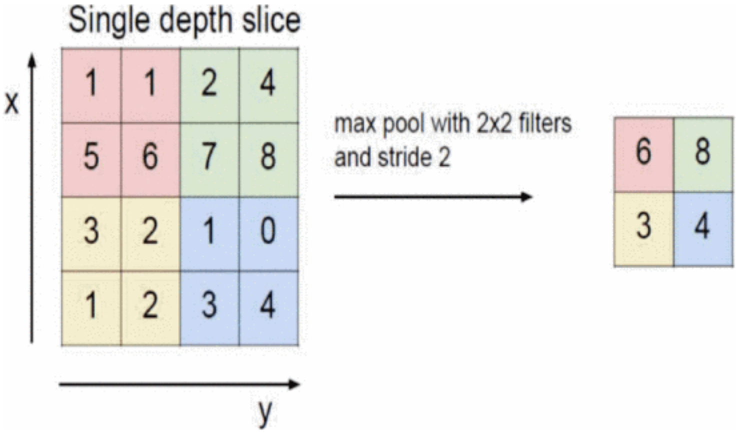


Figure 2.4: Max-pooling with a size of 2 × 2 and a stride of 2 [6].

Chapter 3

Literature Research

The classical approach for determining global aerosol properties or emissions is the data assimilation approach, seen in the previously mentioned paper from Tsikerdekis et al. [4]. As discussed in Section 1.3, Tsikerdekis et al. used an ensemble Kalmann smoother to estimate aerosol emissions from SPEXone measurements. Furthermore, Observing System Simulation Experiments (OSSEs) were performed to obtain synthetic SPEXone observations. The data assimilation used 32 ensemble members. An ideal sensor, named SUPER, in terms of spatial coverage was assumed to act as a benchmark and test the data assimilation system. All experiments were taken in for the period of 20/07/2006 to 20/09/2006. The difference between the results from the data assimilation and the truth (NAT), approached 0 after the data assimilation ran for 26 days. The differences for AOD, AE and AAOD (opposite of SSA) of the data assimilation experiments of SPEXone (or SPX as denoted in the paper) and SUPER with NAT were computed as well as the difference between NAT and CTL. Tsikerdekis et al. found that in both data assimilation experiments an improvement over CTL was present with a global mean error close to zero (see Figure 3.1). Furthermore, they found that SPEXone performed similarly to SUPER, suggesting that the spatial coverage of SPEXone is sufficient to constrain the emissions similar to SUPER. Moreover, the emissions of both data assimilation experiments were improved over the CTL emissions. This is shown in Figure 3.2 and Figure 3.3. The data assimilation experiments were able to constrain the overestimated DU CTL emissions. Only in the western part of the Sahara the experiments had a high ME due to high emissions in that area. Also for the sea salt emissions were the data assimilation results better than CTL. Only over the Indian Ocean high errors occurred for SPEXone due to limited coverage of SPEXone measurements over the region. For both organic carbon and black carbon, the ME and relative ME over high emission sources reached almost zero. Lastly, SO_2 + SO_4 emissions were constrained reasonably well. Tsikerdekis et al. also investigated the impact of a new nature run, with altered emission inventories and emission schemers, to investigate whether the creation of the nature run represented a too simplistic difference between the nature run and data assimilation run. Again they found that data assimilation reduced global emission errors of all species except dust at some locations.

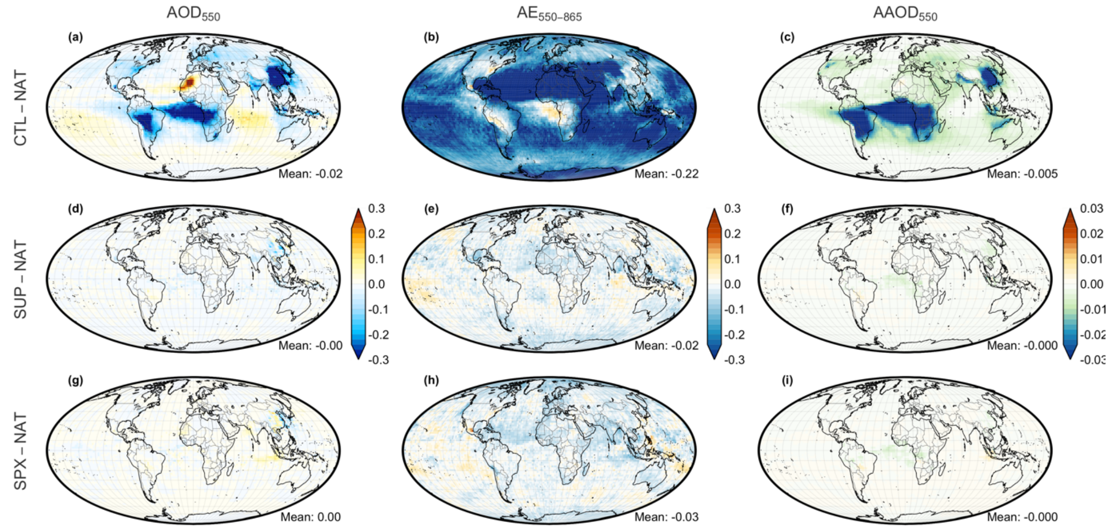


Figure 3.1: Aerosol optical properties differences of CTL – NAT (a,b,c), SUP – NAT (d,e,f) and SPX – NAT (g,h,i). Left column depicts AOD (a,d), middle column AE (b,e) and right column AAOD (c,f). Figure and caption taken from Tsikerdekis et al. [4].

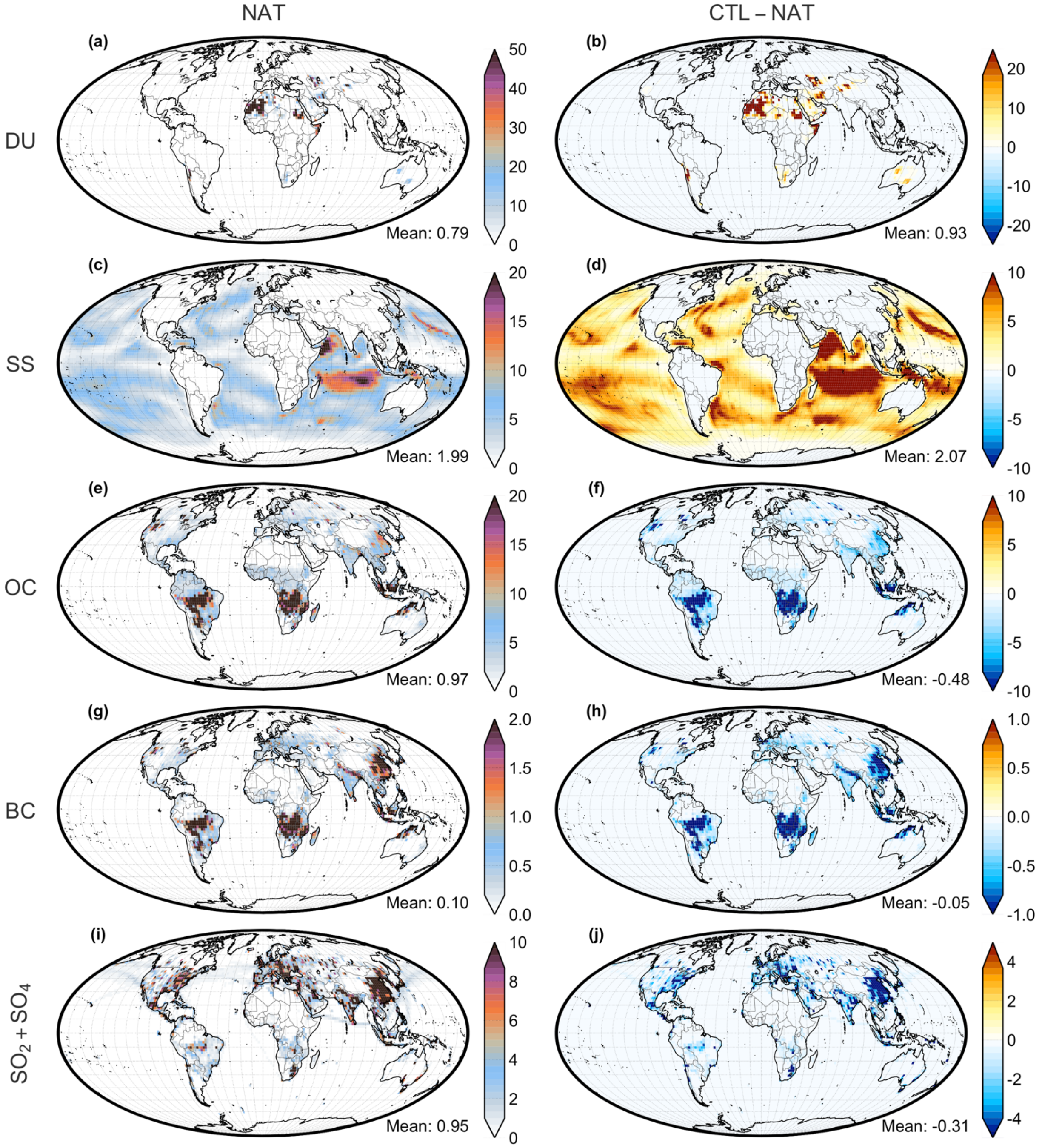


Figure 3.2: Aerosol emission fluxes (kg km-1 day-1) for NAT by species (a) DU, (b) SS, (c) OC, (d) BC, (e) SO2+SO4. The second column depicts the differences between CTL – NAT. Figure and caption taken from Tsikerdekis et al. [4].

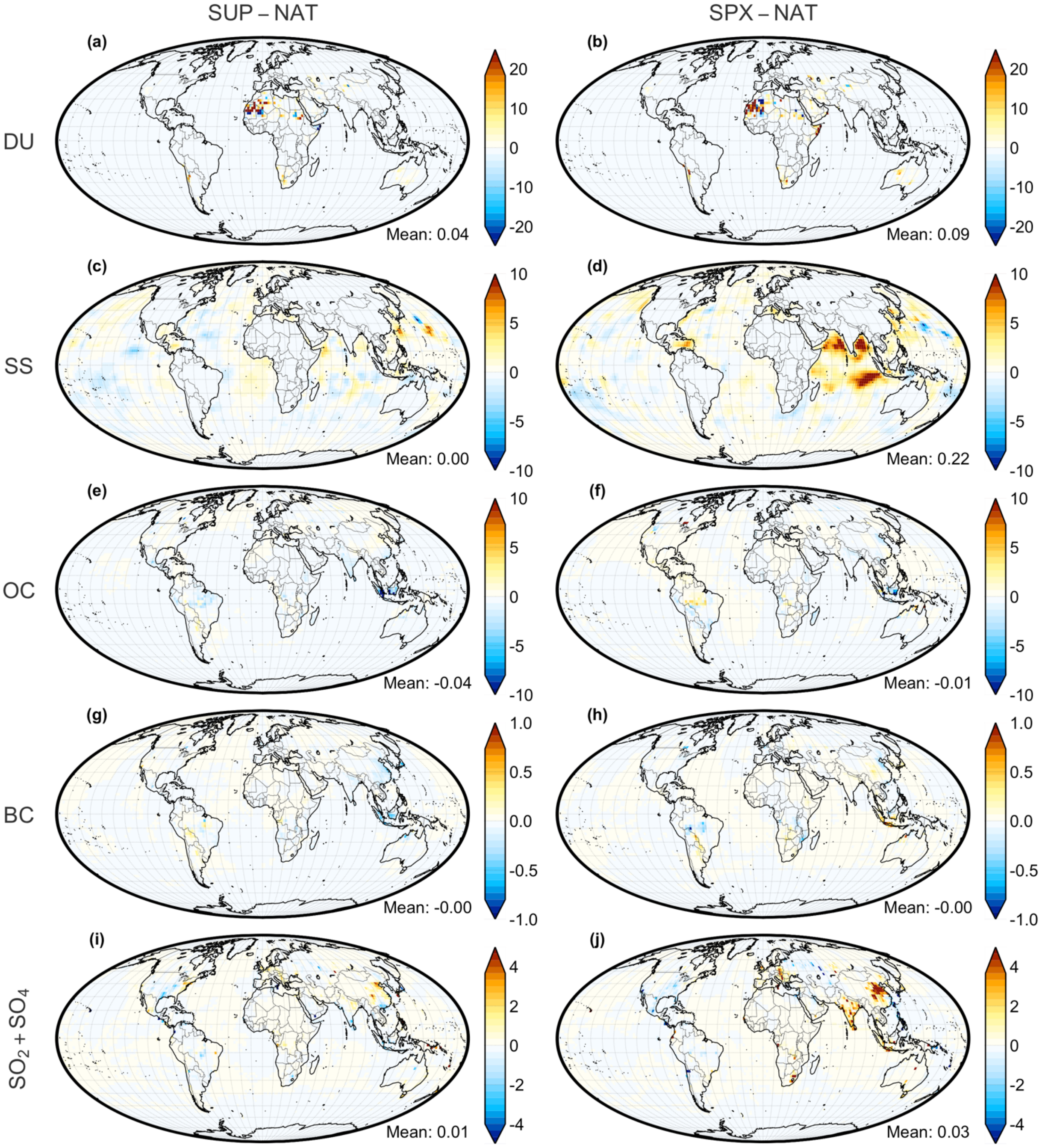


Figure 3.3: Same as Figure 3.2 but for the differences between SUP − NAT and SPX − NAT. Taken from Tsikerdekis et al. [4].

Next we will discuss some papers regarding deep learning.

Machine learning, and deep learning in particular, has become more prevalent in various research fields over the past years. Typical applications of neural networks, such as computer vision, speech recognition and control systems, have similarities to problems in the Earth sciences according to Reichstein et al. [8]. They draw similarities between photos using the RGB values and two-dimensional fields containing data from various variables. Likewise, videos can be compared to a sequence of two-dimensional fields that evolve over time and languages and speech signals share similar characteristics of time series of Earth system variables. Lastly, they argue that classification, regression, anomaly detection and dynamic modelling are all common in both computer vision and Earth science problems.

There are however some challenges for deep learning in the geosciences which need to be overcome. While typical photos have only three channels, namely RGB, some problems in the Earth sciences may have hundreds of channels, which also are correlated to each other. In addition to this, available data from satellites is often noisy or even has missing data and gaps in the fields. Furthermore, for many problems, not a lot of data is available, resulting in difficulties for the training of a neural network. Additionally, deep learning has some challenges not regarding the data. One of these challenges is the often poor interpretability of neural networks, as it is still hard to achieve self-explaining models. Another challenge is physical consistency. Despite the ability to fit the data well, deep learning models may still be physically inconsistent. Lastly, Reichstein et al. argue for integrating deep learning with physical models to obtain better results. An example of a combination of these two models is using deep learning to learn the parameterization of the physical model. Another example is replacing a physical sub-model with a deep learning model. Using this approach, a poorly understood part of a process can be modelled using a neural network and available data, which can then be used in the subsequent parts of the process.

Camps-Valls et al. also discussed the challenging aspects of deep neural networks for geoscience problems and possible ways to integrate them. [9]. They also state the problems of physical inconsistency and poor model interpretability. Additionally, they state that neural networks need to deal with complex data structures, such as multiple outputs, complex statistical relations, different noise sources and high-dimensional spaces.

One example of the application of deep learning in the geosciences is given by Young Seo and Lee [10]. They used satellite data to construct two deep learning models to predict changes in spatiotemporal groundwater storage. A long short term memory (LSTM) model and a combination of a LSTM and a convolutional neural network (CNN) were used and compared to one another. For the training and testing of the networks, monthly observations from 2003 to 2019 on groundwater storage changes, by the National Groundwater Monitoring Network in South Korea, were used. For training, the data from the years 2003 to 2013 were used, and 2013 to 2019 were used for testing. The LSTM-CNN model performed better than just the LSTM, as it had a lower root mean squared error (44.92 mm/month compared to 47.44 mm/month) and a higher correlation coefficient and index of agreement (0.70 and 0.81 compared to 0.62 and 0.77 respectively). Also when the groundwater storage changes were predicted for certain areas in South Korea, the LSTM-CNN appeared to perform better overall than just the LSTM.

As data assimilation techniques are popular and widely used in the geosciences, it is also interesting to analyze the use of machine learning in conjunction with data assimilation.

One of such approaches is considered by Zhu et al., where they suggested a methodology where a neural network is combined with data assimilation to obtain higher accuracy [11]. According to Zhu et al., neural networks are highly capable of approximating complex linear and nonlinear systems, but they lack the physical meanings. Therefore, they introduced a neural network to be trained on the results from a data assimilation approach. This provided an innovation in sensitivity analysis techniques to determine to important input parameters for the neural network. Zhu et al. achieved this with the help of simple fully connected neural networks with one hidden layer and one output layer. First they ran a data assimilation model $F()$ for the first M steps. This produced the results $x_1^{\mathrm{DA}}, \dots x_M^{\mathrm{DA}}$. These results were then used as output data for training the neural network along with the input data $F(x_0^{\mathrm{DA}}) \dots F(x_{M-1}^{\mathrm{DA}})$. The data assimilation model is then updated with the trained network $\mathcal{G}_1$, to obtain the new model $\mathcal{G}_1 \cdot F()$. In the next step, this model will then be used for the next M steps to also train a new network $\mathcal{G}_2$, which will again be used to update the data assimilation model to $\mathcal{G}_2 \cdot \mathcal{G}_1 \cdot F()$. These steps are repeated until a stopping criterion is met or the maximum number of steps is reached. In Figure 3.4 a schematic of this procedure is shown.

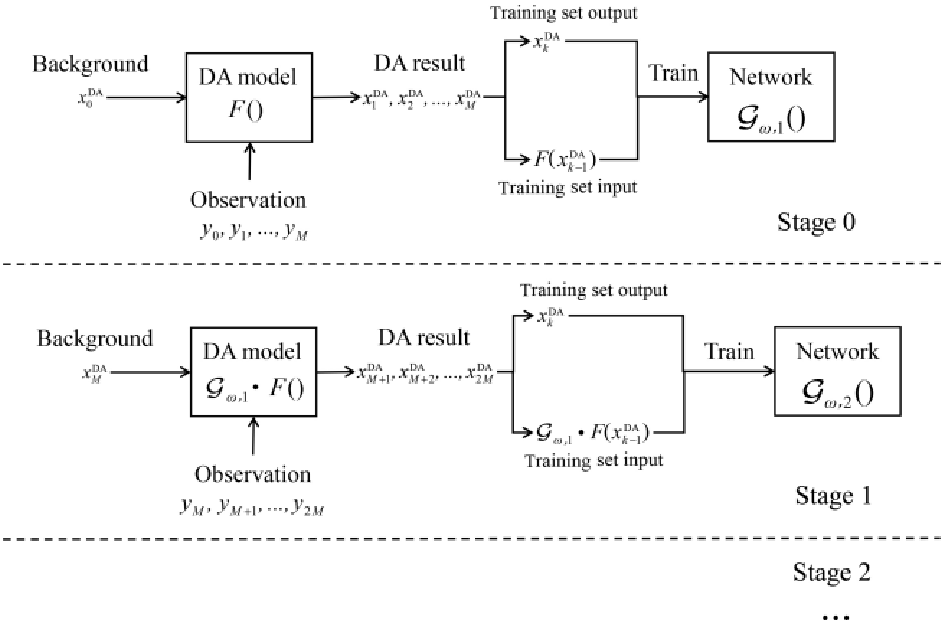


Figure 3.4: A schematic of the used method from Zhu et al. [11].

Zhu et al. applied this method to two problems, a double-integral mass dot system and a Lorenz system. For the double-integral problem, they showed that for the numerical examples, the combined model of data assimilation and neural networks, outperformed the singular model of only data assimilation. It was shown that this was already the case after training the network once, and the performance improved even further after the second training round. For the Lorenz problem, they had the model create forecasting results. Also in this case, the model with the neural network had a better performance, since the model without a network had large forecasting errors, while the model with the network was able to predict the real values quite well.

Brajard et al. also combined data assimilation with deep learning to obtain a model from sparse and noisy observations [12]. They also applied data assimilation to obtain spatially complete data, which is then used as a training set to train the neural network. This trained network is then again used to update the model. For the data assimilation step they used an ensemble Kalmann filter and for the deep learning step a convolutional neural network is used. Brajard et al. performed numerical experiments on the chaotic 40-variables Lorenz 96 model. To assess the capabilities of the model, it was tested on three scores. First, the model's ability to interpolate and de-noise the observational field was tested. This was tested by comparing the spatiotemporal root mean square error (RMSE) of the model with the neural network against the RMSE of the observational field obtained by applying cubic interpolation in space and time and the RMSE of the field obtained from applying the Kalmann filter to the true Lorenz 96 model. They found that the deep learning model reduced the RMSE of the cubic interpolation by a factor of three. The error was however higher than the error of the combination of data assimilation and the true model. However, this was expected, since the observed data were sparse and noisy. In addition, the true model output was not shown to the neural network. Secondly, Brajard et al. compared the dynamical features of the deep learning assisted model to the features of the true Lorenz 96 model. They found that the model was able to reproduce the power spectral density quite well for low frequency's (up to 5 Hz). After 5 Hz, the errors became larger, and there was no significant improvement. Furthermore, the positive Lyapunov exponents of the true model were also well represented by the model with the neural network. Lastly, the forecasting skills of the model were evaluated. The model was simulated for 8 Lyapunov times and compared to the results of the true model. The model was able to forecast the values well for the first 2 Lyapunov times, after which it started to diverge from the true value. After about 5 Lyapunov times, the error started to saturate (see Figure 3.5). Brajard et al. did state that one drawback of the model is its high computational cost, although they did not try to optimize the computational cost.

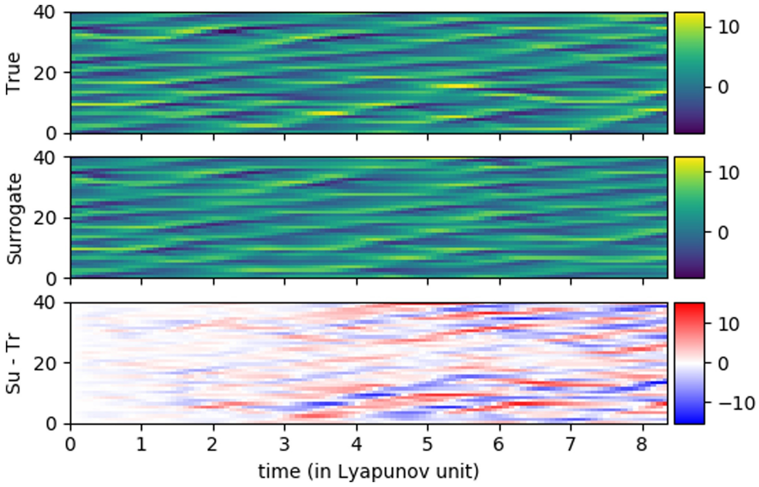


Figure 3.5: Hovmøller plot of one trajectory of the true model, the surrogate model and their difference given the same initial condition with respect to the lead time (in Lyapunov time). 50% of the field is observed with a noise level $\sigma^{obs} = 1$. Figure and caption taken from Brajard et al. [12].

A typical problem used in classic deeplearning tasks with similarities to our problem is image inpainting. For such a problem, the network takes images with missing parts, for example due to file corruption, and tries to fill these missing parts such that the complete image seems realistic. Earlier renditions of this kind of problem only had images with rectangular missing parts, often in the center of the image. Since our missing satellite data does not have such a regular shape, the methods used in those problems would not work for our problem. However, more recently a paper on image inpainting for irregular holes was written by Liu et al. [13]. Liu et al. used a partial convolutional neural network to deal with the irregular shapes. In addition to the partial convolutions, they also implemented an automated update step for the mask for the missing data. The difference between a partial convolutional layer and a regular convolutional layer is that this partial convolutional layer weights the feature update step with the number of unmasked inputs. That is, let $\mathbb{W}$ be the matrix containing the convolution filter weights from the regular convolution step and b its bias. Furthermore, let $\boldsymbol{X}$ be the feature values for the current convolution window and $\boldsymbol{M}$ the binary masks where a 0 corresponds to a missing pixel value. Then the partial convolutional layer updates the feature values as follows:

$$x' = \begin{cases} \boldsymbol{W}^T(\boldsymbol{X} \odot \boldsymbol{M})\frac{\text{sum}(\mathbf{1})}{\text{sum}(\boldsymbol{M})} + b, & \text{if sum}(\boldsymbol{M}) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

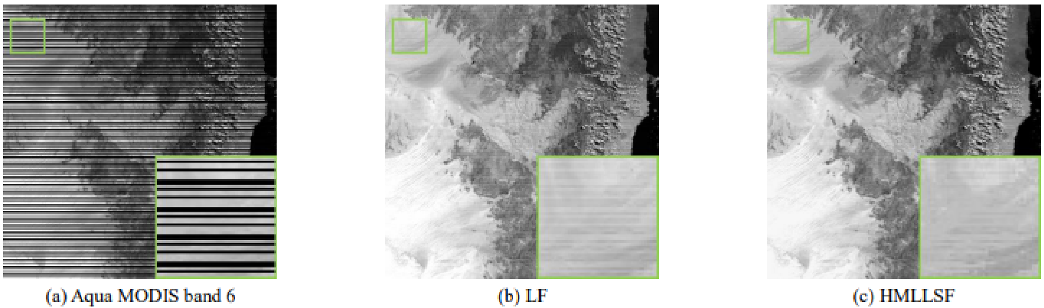
where $\odot$ denotes element-wise multiplication and $\mathbf{1}$ is a matrix with the same shape as $\boldsymbol{M}$ containing only ones. After this step, the mask is updated as follows:

$$m' = \begin{cases} 1, & \text{if sum}(\boldsymbol{M} > 0) \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

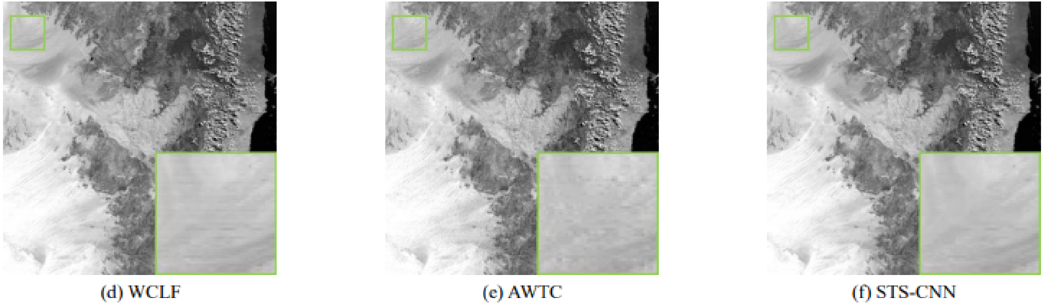
The update in (3.2) shows that the mask values get updated to 1 if the filter was conditioned on at least one non-missing value. The model itself had a U-Net architecture, which makes use of connections between layers in the encoder part and layers in the decoder part of the network (see Ronneberger et al. [14]). For the loss, a custom function was constructed. This custom loss function targets the per pixel reconstruction accuracy as well as the smoothness of the pixel transition into the surroundings. To train the network, more than 55000 randomly generated masks were used. For testing, a little less than 25000 generated masks were used. Liu et al. validated the trained model by comparing their model with four other models on 12000 images with random missing parts in the images. To assess the actual performance of the model compared to the other four models, four different measurements, like the l_1 error and PSNR, were computed. They found that the partial convolutional network outperformed all the other models on all these measurements. In addition to this, Liu et al. conducted a user study, where users were asked which image looked more natural. It was found that in most cases, their model was preferred over the other models. Lastly, Liu et al. note that the model does have two shortcomings. Namely, when the image contains sparsely structured objects, like bars, and when a large amount of data is missing from the image.

The following two papers highlight the combination of a neural network with satellite images to reconstruct missing areas of an image.

Zhang et al. trained a neural network to reconstruct satellite images with missing data [15]. They made a distinction between three problem types: spatial problems, spectral problems and temporal problems. In spatial problems, it is assumed that the missing data has the same distribution as the available data in the surrounding regions, for example as in the problem of Liu et al. [13]. Spectral problems on the other hand, are solved by using spectral data, for example another band of a satellite instrument which is highly correlated to the field with missing regions. Lastly, in temporal problems, temporal data like satellite images of the same region on different days, is used to fill in to reconstruct the image. Zhang et al. proposed to train a CNN, using spatial, spectral and temporal data to reconstruct the image. Instead of regular convolutions, dilated convolutions were used. Diluted convolutions have bigger filters, allowing them to cover more of the images, while still using the same amount of non-zero elements in those filters. This allows the network to use more contextual information, like a bigger filter size in a traditional convolution layer, while keeping the computational cost low. In addition to diluted convolutions, skip connections were used to maintain more information in the later stages of the network. The network was trained on 600 images of the Terra MODIS satellite, divided into smaller images. The model then was tested on images from the Aqua MODIS band 6 against a multitude of other methods. All methods performed well, but when inspecting an enlarged area of the image, some noise was still present for the other methods, while it was not apparent for the model from Zhang et al. This can be observed from Figure 3.6



(a) Aqua MODIS band 6 (b) LF (c) HMLLSF



(d) WCLF (e) AWTC (f) STS-CNN

Figure 3.6: Real recovery results for Terra MODIS band 6 taken from Zhang et al. [15]. Figure (f) shows the results from the trained network from Zhang et al. (b) - (e) are the results from other methods.

A paper which is closely related to our problem is the paper by Lops et al. [16]. In the paper, they trained a Partial Convolutional Neural Network (pCNN) to construct missing regions from an AOD field observed by a satellite. They used satellite data from the Geostationary Ocean Color Imager (GOCI) instrument to obtain missing data masks to be used for training. These images had missing data between 35% and 99.7% with approximately 45% of the images missing 80% or more data. These masks were applied to results from the Community Multiscale Air Quality (CMAQ v5.2) model to obtain the input images and the target images for training. As for the neural network, Lops et al. implemented partial convolutions with a U-Net architecture. Furthermore, the custom loss function as seen in [13] was used. To evaluate the network, daily mean data of the GOCI images was used. Various metrics, such as the MAE, RMSE, correlation coefficient (r) and Index of Agreement (IOA), were used and compared to other models and algorithms. It was found that the pCNN achieved average results with respect to the IOA and r, while it obtained the second lowest MAE and RMSE. However, when observing the results for cases depending on the percentage of missing values in the images, the pCNN started to outperform the other models, when the images exceeded 60% missing data. These results are shown in Figure 3.7. In addition to these metrics, the bias of individual filled in pixels was calculated for various distances from this pixel to the nearest pixel with available data. This was again compared to other models. Lops et al. showed that for smaller pixel distances, so the pixel which was predicted was close to available data, the pCNN performed similarly to the other models. For larger pixel distances however, the pCNN had a negative bias with a smaller bias variance compared to a positive bias with higher variance of the other models. A shortcoming of the model however, is that the network was not able to properly predict certain cases where there were clusters of available data near the borders of the images. This resulted in poor prediction from the model for pixels on the other side of the image from where the data was available.

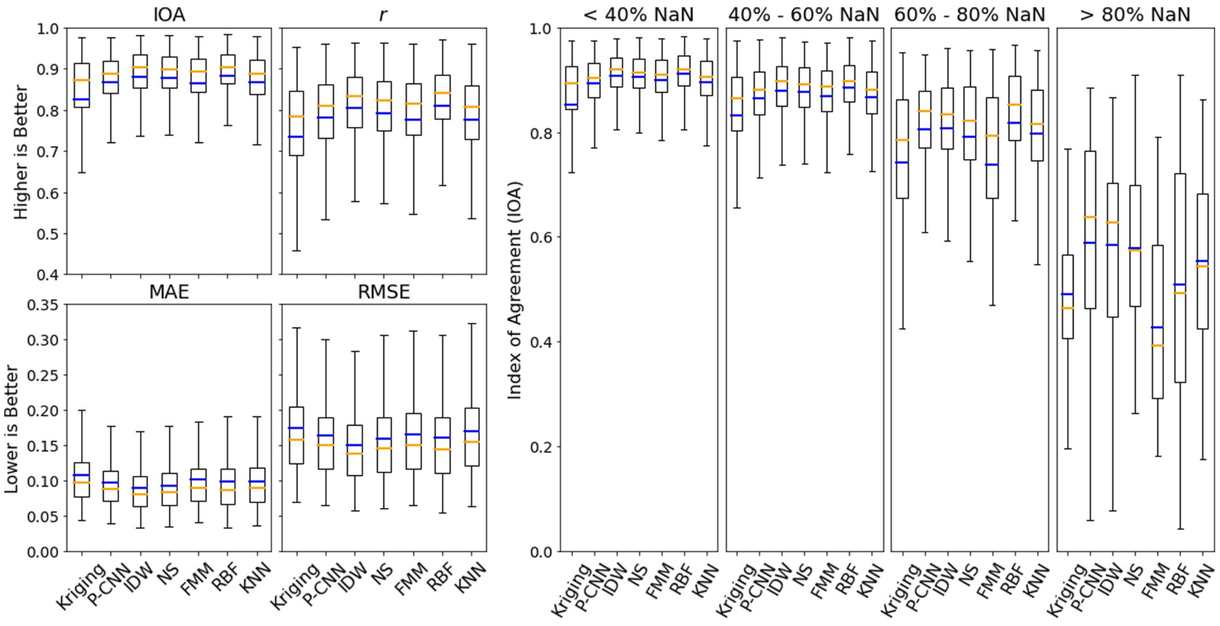


Figure 3.7: Statistical comparison of the imputed missing Geostationary Ocean Color Imager data from the hourly mask to the daily mean mask. The left figures indicate the statistical performances of the partial convolutional neural network (PCNN) model, Kriging (Kriging), inverse distance weighted (IDW), Navier Stokes (NS), fast marching method (FMM), radial basis function (RBF), and K-nearest neighbor (KNN) methods. The evaluations are based on the index of agreement (IOA), the correlation coefficient (r), the mean absolute error (MAE), and the root mean squared error (RMSE). The right figure indicates the IOA performance of the models, based on the percentage of missing data split into three categories. The main section of the boxplot presents the interquartile range between the 25th and 75th percentiles. The whiskers (vertical lines) of the boxplot represent the variability outside the interquartile range. The blue and yellow horizontal lines represent the mean and median of the data set, respectively. Figure and caption taken from Lops et al. [16].

Chapter 4

Datasets

This chapter will provide more information on the datasets used for training the neural networks, such as the input and output variables. Section 4.1 contains information on the dataset for the aersol properties problem, while Section 4.2 provides information on the dataset for the emissions problem.

4.1 Aerosol Properties Dataset

The dataset for the first problem contains information about the AOD, SSA and AE. As discussed in Section 1.3, the SPEXone measurement instrument has not been launched yet, therefore no real life data from the instrument were used. Instead, a run of the ECHAM-HAM model was considered as the truth of the aerosol properties. The ECHAM-HAM simulations are for the year 2006 with a horizontal resolution of 1.875×1.875 degrees. For the targets of the neural network, from now on referred to as the nature run (NAT), were results from a data assimilation experiment using measurements from the POLDER instrument, also of the year 2006. Each sample of the NAT run is the global field of one day for the AOD, SSA and AE respectively. For each of those days a SPEXone mask was applied to the NAT field, to represent what would be observed in that day by the instrument. Due to the limited coverage of between 3 and 6 percent of the globe per day, there are a lot of missing values after applying the mask. In addition to these SPEXone fields, the results of another model simulation were used to assist the model with dealing with the missing regions. The results of this run, from now on the control run (CTL), were obtained with the same model, but with altered emissions, to get different results from the NAT run. These three fields, that is the NAT field, the SPEXone field and the CTL field, have the same shape of 96×192, where the coordinates are in $(latitude, longitude)$. Lastly, in addition to the CTL field, the gradient fields of the CTL field in both the latitude and longitude direction are included, to provide more information on the transport of the aerosols.

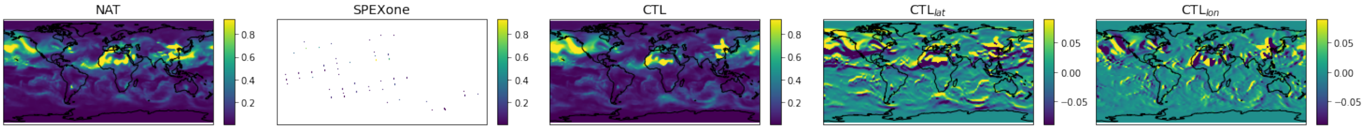
The data fields and their shapes are shown in Table 4.1. Note that these data fields exist for AOD, SSA and AE with different values. An example of a sample for AOD, SSA and AE is shown in Figure 4.1. The figure demonstrates that, while the NAT field and the CTL field look similar, there are also clear differences between them. The training and validation datasets for the model consist of 365 days each of the same year. The NAT fields of the training dataset were constructed with the same model, but with different emissions. The NAT run from the training dataset is slightly correlated to the NAT run from the validation set. However, the difference is big enough for the network to be unable to predict the validation field by replicating the predicted field during training. The CTL run was the same for both datasets. The SPEXone map was different between the sets, due to the difference of the NAT runs. A sample from the training and testing dataset is shown in Figure 4.2 for AOD, SSA and AE. To further increase the amount of training data, some augmented data from the training set was added. The original CTL, SPEXone and NAT images were flipped horizontally, vertically and both horizontally and vertically and then added to the original dataset. While the flipped images are technically the same, the network does not recognize them and sees them as entirely new data. This results in four times the data while also having more diverse data. Figure 4.3 shows an original sample of a AOD CTL field along with the same image flipped horizontally, vertically and both horizontally and vertically. This is a form of data augmentation, which regularizes the model and improves generalization, which in turn reduces overfitting [17].

To be able to apply a convolutional filter to the images in the CNN, an extra dimension has to be added to the 2D fields, often called the channel axis. In typical images this axis has three channels, namely for the RGB values of an image. However in our case, while we do have three different modes to train in the AOD, SSA and AE values, we only have one channel, since these modes were trained separately. AOD, SSA and AE were trained separately since it was found to have better results. With the added axis, the new dimensions of the fields are $96 \times 192 \times 1$.

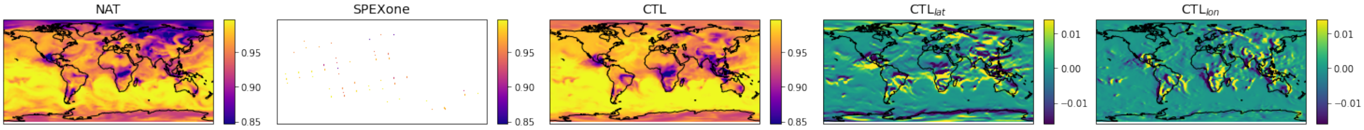
Lastly, since some of the aerosols in the atmosphere on a day can still be present in the atmosphere one week later [18], the values of the fields of a day are correlated to previous and next days. Therefore, some information can be taken from the days around the sampled day. To feed this information to the network, the SPEXone measurements of the previous and the next day were also included for each day. These days were added along the channel axis for the SPEXone field, resulting in a new shape of $96 \times 192 \times 3$. As a result, the original dataset now consists of 363 days, since the first and the last day of the year do not have an available previous and next day respectively. With the flipped images included, the final dataset used for training had 1452 samples.

Input			**Output**		
data field	full name	shape	data field	full name	shape
SPEXone	SPEXone measurements	96×192	NAT	Nature run	96×192
CTL	Control run	96×192			
CTL_{lat}	Control run gradient latitude direction	96×192			
CTL_{lon}	Control run gradient longitude direction	96×192			

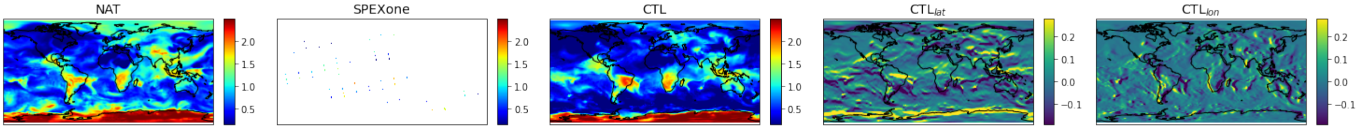
Table 4.1: The input and output data fields that were used for the aerosol properties problem. These fields are the same for AOD, SSA and AE, except for the actual values of the fields.



(a) AOD

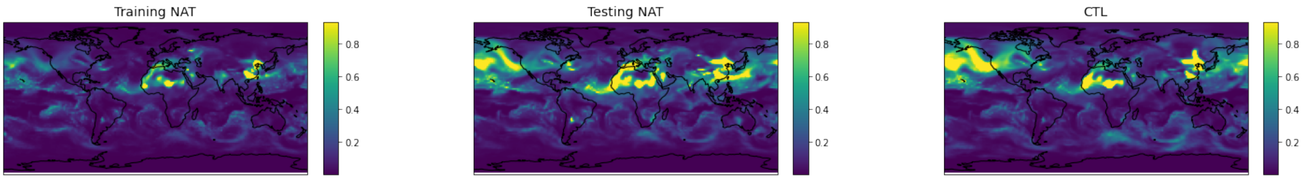


(b) SSA

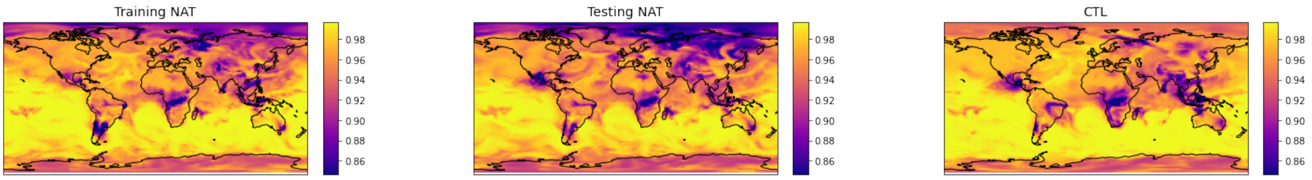


(c) AE

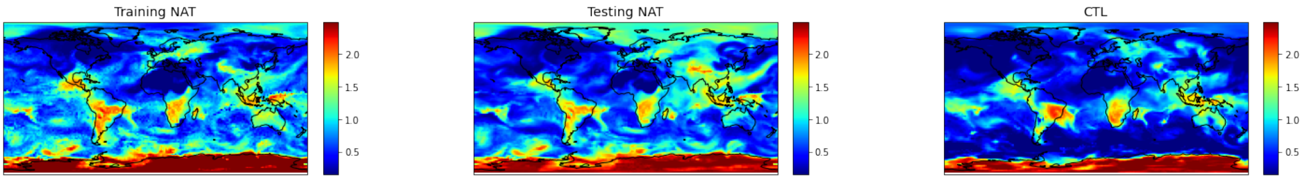
Figure 4.1: Sample day from the dataset for AOD, SSA and AE. Note that the coverage of SPEXone is the same for all three modes, since the same day was taken. The values in the covered SPEXone, however, do differ.



(a) AOD



(b) SSA



(c) AE

Figure 4.2: AOD, SSA and AE field for a sample day from the training dataset and the testing dataset. Note that the CTL field is the same for both datasets, hence it is only included once.

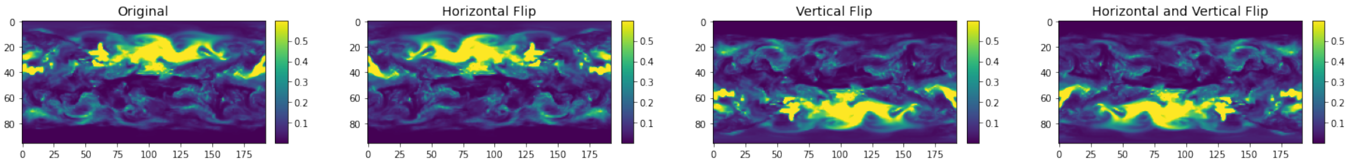


Figure 4.3: An original CTL field of AOD as well as the field flipped horizontally, vertically and both horizontally and vertically.

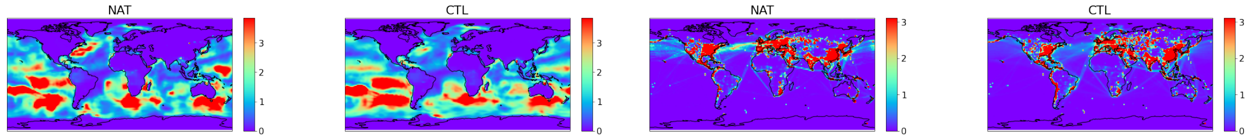
To prepare the data for feeding it to the network, some alterations were made to enhance training performance. Firstly, since a neural network cannot work with the missing values from the SPEXone fields, these values were changed to -1. This value was chosen for the reason that the AOD, SSA and AE have a minimimum of 0, therefore -1 still represents an unknown value. Secondly, the CTL input variables were re-scaled to be in the interval $[0, 1]$. The SPEXone inputs, however, were not normalised, since the known pixels of the middle day contain the correct values from the output, while the rest is equal to -1. Thirdly, we took the logarithm of the AOD and SSA fields. This was done as it improved the smoothness of the neural network output, due to it having an easier time with predicting values which lie further apart. Lastly, the dataset was divided into batches with 32 samples each, since a neural network trains on batches of the dataset to be able to find a pattern in the batch. Due to the fact that all batches were chosen to be equal size, this left 12 samples which could not be used. Therefore, the final dataset consisted of 1440 samples.

4.2 Emissions Dataset

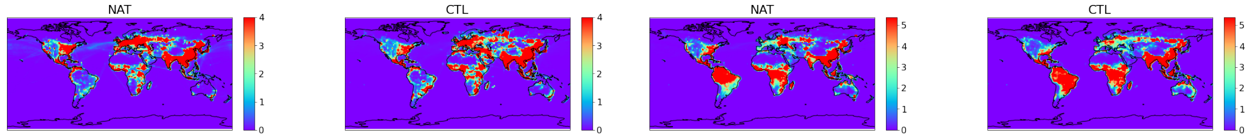
The dataset for the second part of the problem again contains information about the AOD, SSA and AE, since the emissions themselves cannot be measured with SPEXone. However, instead of using the CTL fields and the SPEXone measurements directly as input for the network, they were first used as input for the trained network for aerosol properties (see Section 5.1 for the network and Chapter 6 for its results) to obtain daily globally covered fields with a higher accuracy than CTL. Similar to the CTL fields in Section 4.1, the gradient of these fields were taken in longitude and latitude direction. Therefore these fields can be considered as CTL fields with a higher accuracy. Since the SPEXone measurements are already used to construct these fields, via the first trained network, they were not also used as an input for the network of the emissions problem. Unlike it was the case for the aerosol properties problem, samples for the emissions problem were not single days in a year like in the previous problem. Instead, each sample, that is all input variables and all output variables, was the average of 7 days. So if our first sample would be the average of days 1 to 7, then the next sample is the average of days 2 to 8. This approach was chosen since it was easier for the model to learn as the results were less noisy due to the NAT fields having a smoother transition. In addition to taking the average emissions over 7 days, the emission fields are also scaled for the training process. The emissions of NAT were scaled by dividing them by the global mean of the CTL field for each emission species. These CTL fields of the emissions were not used as input for the network. The emissions were scaled because the emissions were originally in $\frac{kg}{m^2 s}$, therefore they were of the order 10^{-9} or less. These small values are hard for the neural network to learn. As mentioned previously, there are six different emission types, namely dimethylsulfide (DMS), sulphur dioxide (SO_2), black carbon (BC), organic carbon (OC), sea salt (SS) and dust (DU). From these emissions, the types OC, SS and DU have 3, 2 and 2 modes respectively. This results in a total of ten different size and hygroscopicity modes which need to be predicted. Since these modes are correlated to each other if they belong to the same aerosol species, they are not trained separately from each other. Instead, only the different species are separated. If a species contains multiple modes, they are concatenated along the channel axis. This can be interpreted as an RGB image, which has channels for the same image. For example, SS has 2 different modes, therefore the SS output image will be of shape $96 \times 192 \times 2$. Likewise, the information of AOD, SSA and AE are concatenated along the channel axis for each separate input respectively. For example, the AOD input is of shape $96 \times 192 \times 3$, containing the AOD fields from the trained network and its gradient fields along the latitude and longitude direction. The NAT fields from the training dataset were constructed from 8 out of the 32 different ensemble members used for the construction of the original NAT model from data assimilation. Typically, data assimilation uses ensembles to obtain more accurate results. An ensemble is constructed from multiple ensemble members, in this case 32. Each ensemble member has a slightly different initial state, for example by adding a perturbation to some deterministic initial state. These different initial states are then run through the model to obtain 32 different outcomes. Afterwards, the mean is taken over these ensemble members to gain the final results of the data assimilation model. The ensemble members were chosen from the results from the same model used for the NAT fields for training the aerosol properties network. The testing data consisted of another ensemble member of the same run. The CTL fields used to construct the globally covered SPEXone fields were again taken from the results from the same model but with altered emission sources. A sample for the 7 day average NAT field and CTL field is displayed in Figure 4.4 for each emission species. Note that both NAT and CTL are scaled the same, that is by dividing them by the global average of CTL.

Input			**Output**		
data field	full name	shape	data field	full name	shape
AOD	AOD information	$96 \times 192 \times 3$	DMS	Nature run DMS	$96 \times 192 \times 1$
SSA	SSA information	$96 \times 192 \times 3$	SO_2	Nature run SO_2	$96 \times 192 \times 1$
AE	AE information	$96 \times 192 \times 3$	BC	Nature run BC	$96 \times 192 \times 1$
			OC	Nature run OC	$96 \times 192 \times 3$
			SS	Nature run SS	$96 \times 192 \times 2$
			DU	Nature run DU	$96 \times 192 \times 2$

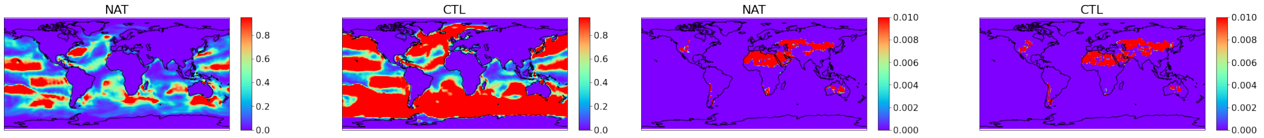
Table 4.2: The input and output fields that were used for the emissions problem.



(a) DMS (b) SO_2



(c) BC (d) OC



(e) SS (f) DU

Figure 4.4: A sample of the NAT field and the CTL field of the 7 day average emissions for DMS, SO_2, BC, OC, SS and DU. Both NAT and CTL are scaled by dividing the original field by the global mean of the CTL field.

Similar to the first problem the data fields were altered to increase performance of the network training. Again, the missing values in SPEXone were replaced with -1 to indicate the lack of available data. Additionally, the AOD, SSA and AE fields and their gradients were also again rescaled to the interval $[0, 1]$. Lastly, the dataset was divided into batches with 16 samples per batch. Another possibility was a batch size of 32 similar to the aerosol properties dataset, since there were no significant differences in performance between 16 and 32.

Chapter 5

Network Architecture

This chapter will describe the choices made for the architecture of the neural networks. First, the architecture for the aerosol properties is explained in Section 5.1 followed by the architecture for the emissions problem in Section 5.2

5.1 Aerosolic Properties Network Architecture

The neural network for the aerosol properties problem has an encoder-decoder structure with two different inputs. In an encoder-decoder structure the inputs go through the decoder, after which the network has learnt a latent representation of the input. This representation will then be decoded through the decoder to obtain the output. This is a standard approach for CNN's with an image input and an image output. The inputs are one input for the CTL field with its gradient in both longitude and latitude direction and one input for the SPEXone fields for the previous, current and next day. This means that both input layers have three channels. The inputs were separated as SPEXone has a different structure from the CTL inputs. Therefore, the network can have an easier time during training. After the input layer, 4 convolutional layers follow. Next down-sampling is performed via the use of a convolutional layer with stride 2, instead of the typical maxpooling. This way, the model can also learn the down-sampling operation instead of it being a fixed one. After the down-sampling, some more convolutional layers are added. This part of the network, the encoder, is the same for both inputs. At the end of the encoders of the inputs, the output of both encoders are concatenated along the filter axis, to be fed into the decoder. The decoder consists of deconvolutional layers as well as one upsampling layer. The upsampling in the decoder is not done by using a stride of 2, as it did not provide any improvement on the performance of the network. The model also has two interconnections from the encoders to the decoder, one before the up- and downsampling and one after, to help reduce information loss through the network. An interconnection is a direct link from one layer in the encoder to another layer further in the model in the decoder. This connection is typically used to provide information to a layer in the decoder, which can be lost due to vanishing gradients through the other layers. This information from the encoders is also concatenated along the filter axis to the layers in the decoder. The model architecture is displayed in Figure 5.1. The orange layers represent convolutional layers, while the blue layers represent the deconvolutional layers. Below each layer, the number of filters in that layer is stated. Each layer uses the ReLU activation function, except for the last layer, for the output, which uses a linear activation function. The output layer has a linear activation function, since it is a regression problem, so all output values are a possibility. Finally, after each convolutional and deconvolutional layer, batch normalization was used. The model architecture was the same for AOD, SSA and AE.

Different architectures were tested, but this architecture proved to be the best performing. For one, we experimented with different depths of the network. If the network contains less layers, it will not be able to learn the problem well, resulting in worse results for both the training data and the testing data. However, if the network has more layers, it will overfit, with slightly better training results but worse validation results as a consequence. The same applies to the number of filters used in the layers, too many filters in the layers resulted in an overfit, while too few caused the model to underfit. Additionally, we tested more and fewer interconnections from the encoder to the decoder, but two skip connections performed the best. Lastly, the number of downsampling and upsampling layers was considered as well as the method used, maxpooling or using a stride of 2. Here we found that no downsampling resulted in a more noisy map, with some parts containing small blocks or lines with over- or underprediction of the values. Likewise, with two or more downsampling layers, the network lost too much information, which lead to an underfit. The choice for using convolutional layers with a stride of 2 instead of maxpooling is that, unlike with CNN classification problems, the fields are smoother without any real big 'objects' with distinct features. Therefore, only taking the maximum in a 4×4 block can lose a lot of information. As mentioned before, for upsampling there was no real difference between using a regular upsampling layer and a deconvolutional layer with stride 2. Thus, we chose for a regular upsampling layer to decrease computational cost.

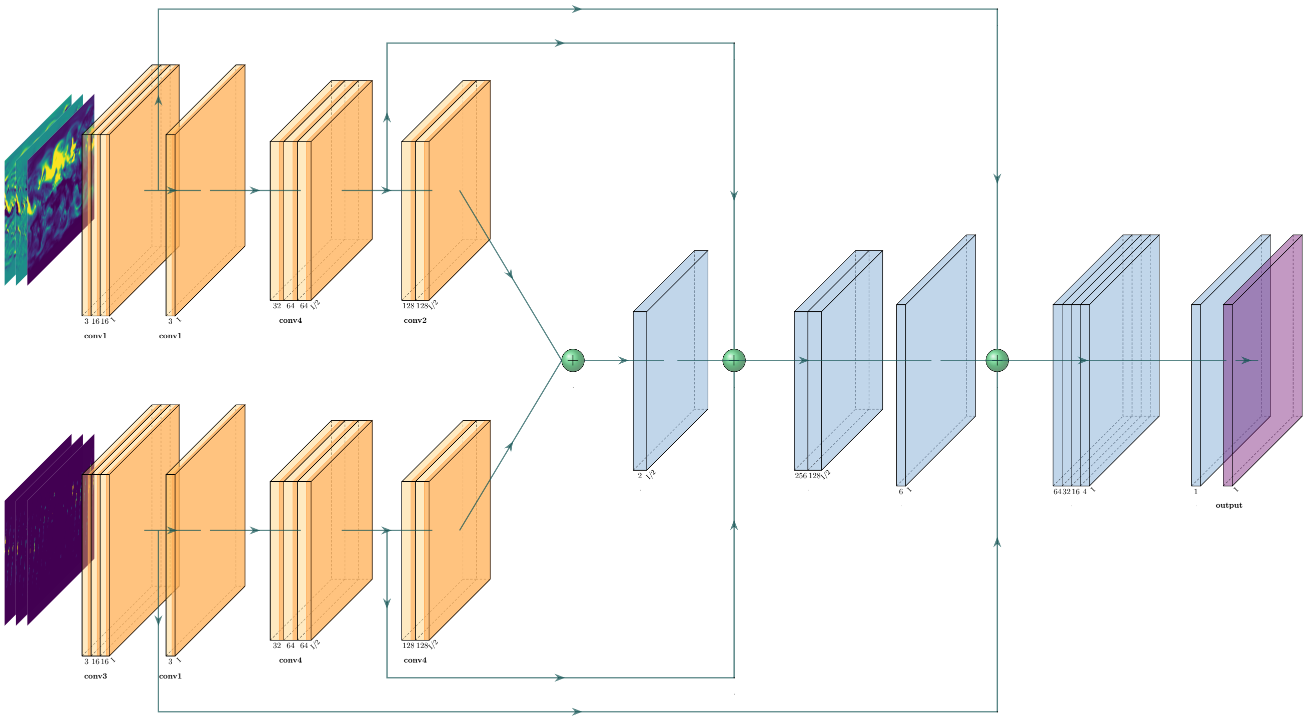


Figure 5.1: The architecture of the network for the aerosolic properties problem.

5.2 Emissions Network Architecture

The neural network for the emissions problem again has an encoder-decoder structure like used for the aerosol properties problem. Unlike the three different networks for the aerosol properties, this problem used one network for all emission species. This approach was chosen because it did not give better results when each species was trained separately, as was the case for the aerosol properties. Therefore, to make it more convenient, only one network was trained, although it came with an increased computational complexity. This model has three inputs, namely AOD, SSA and AE and it has six outputs, DMS, SO_2, OC, BC, SS and DU. The input layers for AOD, SSA and AE all have 3 channels to account for the field containing the actual values and the gradient of the field in latitude and longitude direction. The number of channels in the output layers differ depending on the number of different modes for the species. For example, DMS only has one output channel, while OC has 3 channels. The encoder, of which the structure can be found in Figure 5.2, is structured the same for all three inputs. First, 4 convolutional layers which increase in the number of filters are present. The output of the last convolutional layer of this block is sent to the next block in the encoder and to each decoder through a skip connection. Additionally, down sampling is performed with the use of max pooling. The next block of the encoder consists of only 2 convolutional layers. Similar to the previous block, the output from this block is also sent to each decoder via a skip connection and down sized with max pooling. The last block consists of 3 convolutional layers followed by a dropout layer, with a parameter of 0.2, which means that randomly 20% of the input is set to 0 at each training step. This dropout layer was added to reduce overfitting of the model. The output of the three encoders are summed up, instead of concatenated along the filter axis. They were not concatenated due to the model being larger and more complex. Therefore, adding the outputs helps for computational performance. The sum of the encoders is then fed into each decoder. The structure of each decoder is again the same and is given in Figure 5.3. The decoder starts with 3 deconvolutional layers. After this block, upsampling is performed with the built in function. The output of the second block from the encoder is then added to the upscaled images, followed by 2 deconvolutional layers. Next, the images are again upsampled and added to the output from the first block of the encoder. Lastly, the data is fed to 4 more deconvolutional layers, which leads to the output. Each layer uses the ReLU activation function, except for the output layer, which uses a linear activation function. After each convolutional layer and deconvolutional layer, batch normalization is performed. The architecture of the full model is shown in Figure 5.4, where each orange block represents an encoder with the architecture from Figure 5.2 and each blue block represents a decoder from Figure 5.3. The skip connections were omitted to increase readability.

Again, different architectures were tested. Overall, we found the same results as for the different architectures for the aerosolic properties problem. However, some aspects were different. Firstly, the model was trained for all outputs in the same model instead of a separate model for each output, since it did not improve the results like it did for the aersololic properties. As a consequence, however, the model is more complex. Therefore, downsampling was performed twice to reduce computational cost. Secondly, maxpooling gave the same results as the use of a convolutional layer with a stride of 2. To further reduce computational cost, maxpooling was therefore used. Lastly, due to the problem being more complex, a higher filter size in the later layers of the model was used.

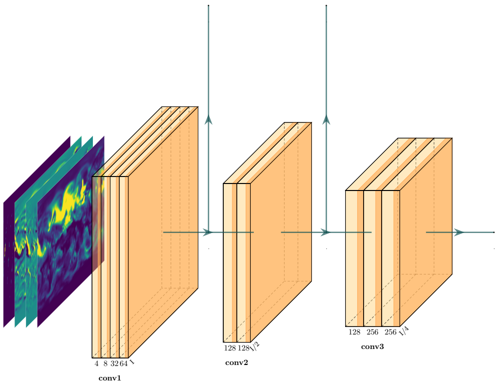


Figure 5.2: The encoder part of the network for the emission problem.

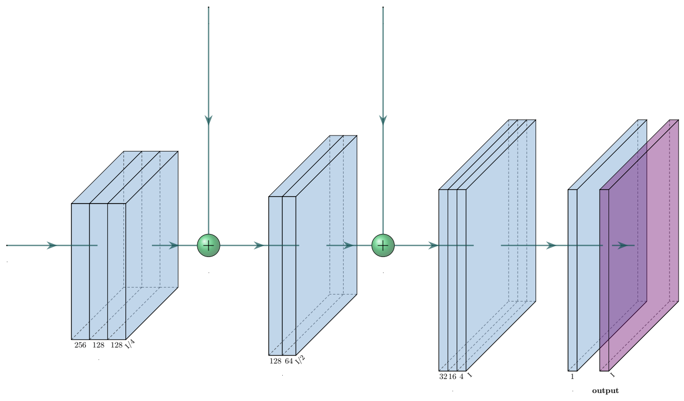


Figure 5.3: The decoder part of the network for the emission problem.

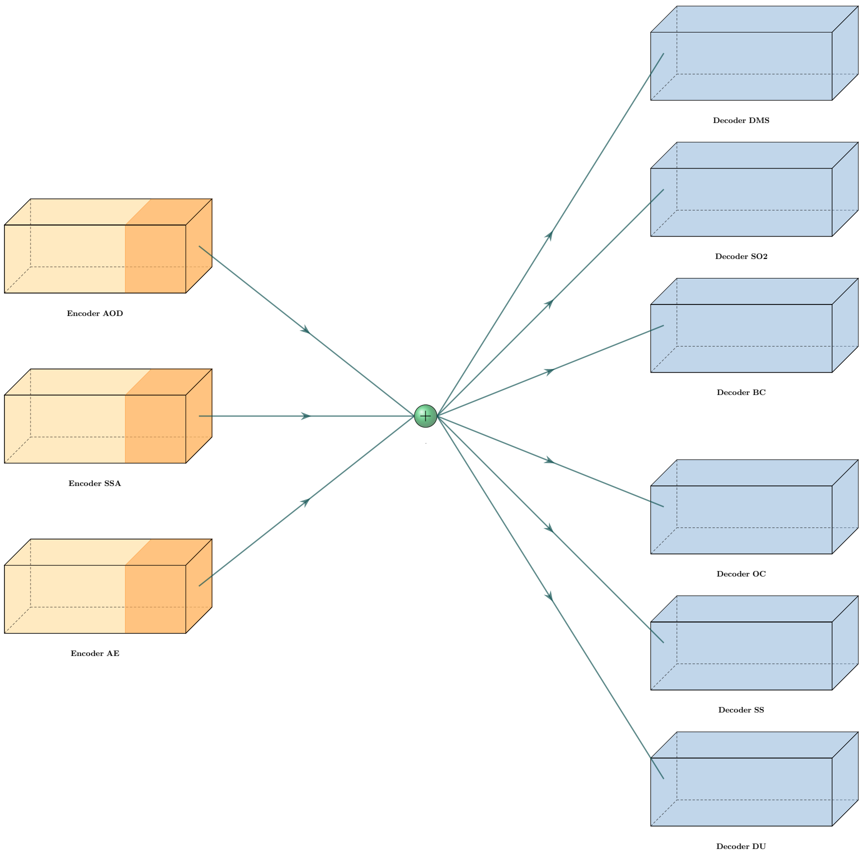


Figure 5.4: The complete architecture of the network for the emission problem. All the encoders have the structure from Figure 5.2 and all the decoders have the structure from Figure 5.3

Chapter 6

Results for the Aerosol Properties Problem

This chapter will present the results from the trained neural network for the aerosol properties problem. First, the results of the training progression are provided in Section 6.1. Secondly, the results from the model are discussed in Section 6.2. Thirdly, the influence of the inclusion of SPEXone measurements are investigated in Section 6.3. Lastly, in Section 6.4 the results of two other performed experiments are discussed.

6.1 Training Progression

The network was trained on Kaggle, a subsidiary of Google which provides the use of GPU's for free[1]. The model ran for 200 epochs with a batch size of 32. These parameters were chosen by trial and error. Furthermore, the mean absolute error was used as the cost function. The Adam optimizer from Keras was used with a learning rate of $5 \cdot 10^{-4}$ and a decay of $\frac{5 \cdot 10^{-4}}{200}$. The training progression for the models of AOD, SSA and AE is shown in Figure 6.1. From the figures, we can see that the model converges in all three cases. The training loss after the final epoch is 0.11, 0.0063, and 0.067 for AOD, SSA and AE respectively, while the validation losses are 0.21, 0.014, and 0.1. This relatively small difference between training losses and validation losses implies that the model has been able to learn a mapping from the input to the output, instead of finding a structure in just the training data. While the validation loss of the SSA has a big spike, the overall losses are still small. To evaluate the performance of the model further, some tests are done using the validation data.

(a) AOD (b) SSA (c) AE

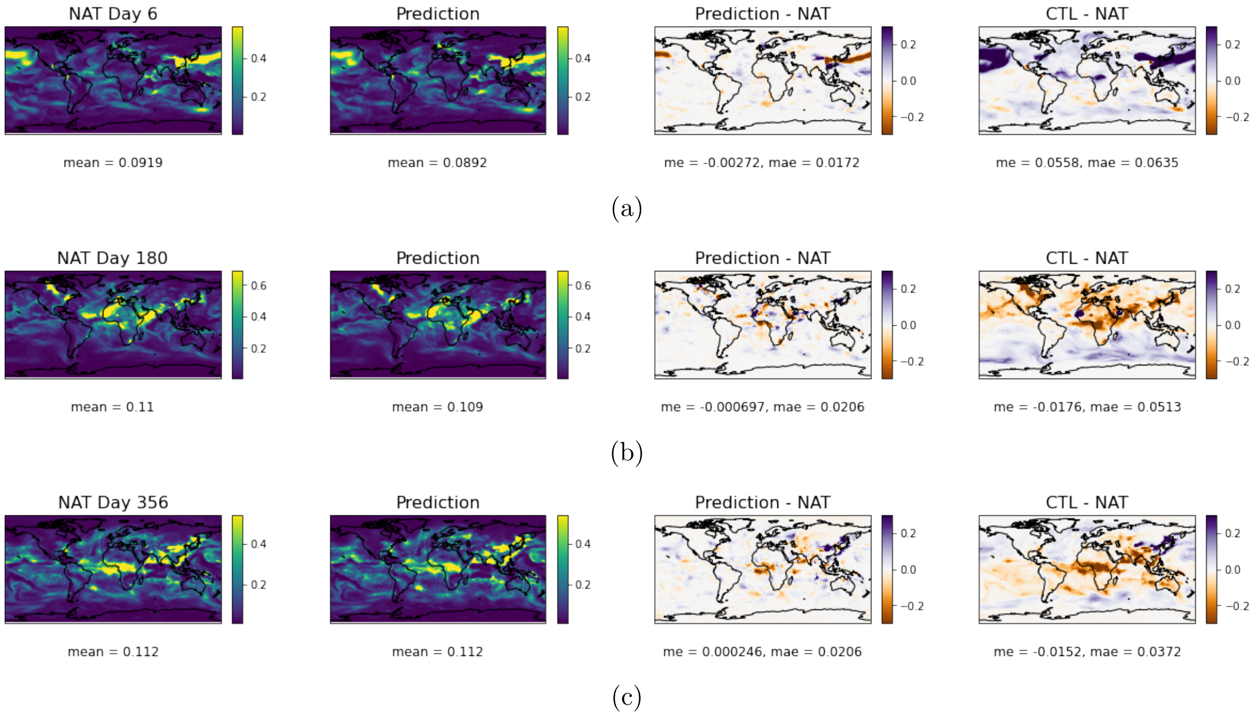
Figure 6.1: Training loss and validation loss against the epochs for the models for AOD, SSA and AE. The loss function used is MAE.

---

[1] https://www.kaggle.com

6.2 Model Predictions

Several predictions were made on the validation data. Furthermore, different errors and plots were made on those predictions to analyze the results. Firstly, we chose three different days throughout the year, one at the start, one at the middle and one near the end of the year. Both the results of the model and the control run were compared to the nature run to investigate whether there would be an improvement over the control run and if so, how big of an improvement that would be.


NAT Day 6 | Prediction | Prediction - NAT | CTL - NAT

mean = 0.0919 | mean = 0.0892 | me = -0.00272, mae = 0.0172 | me = 0.0558, mae = 0.0635

(a)

NAT Day 180 | Prediction | Prediction - NAT | CTL - NAT

mean = 0.11 | mean = 0.109 | me = -0.000697, mae = 0.0206 | me = -0.0176, mae = 0.0513

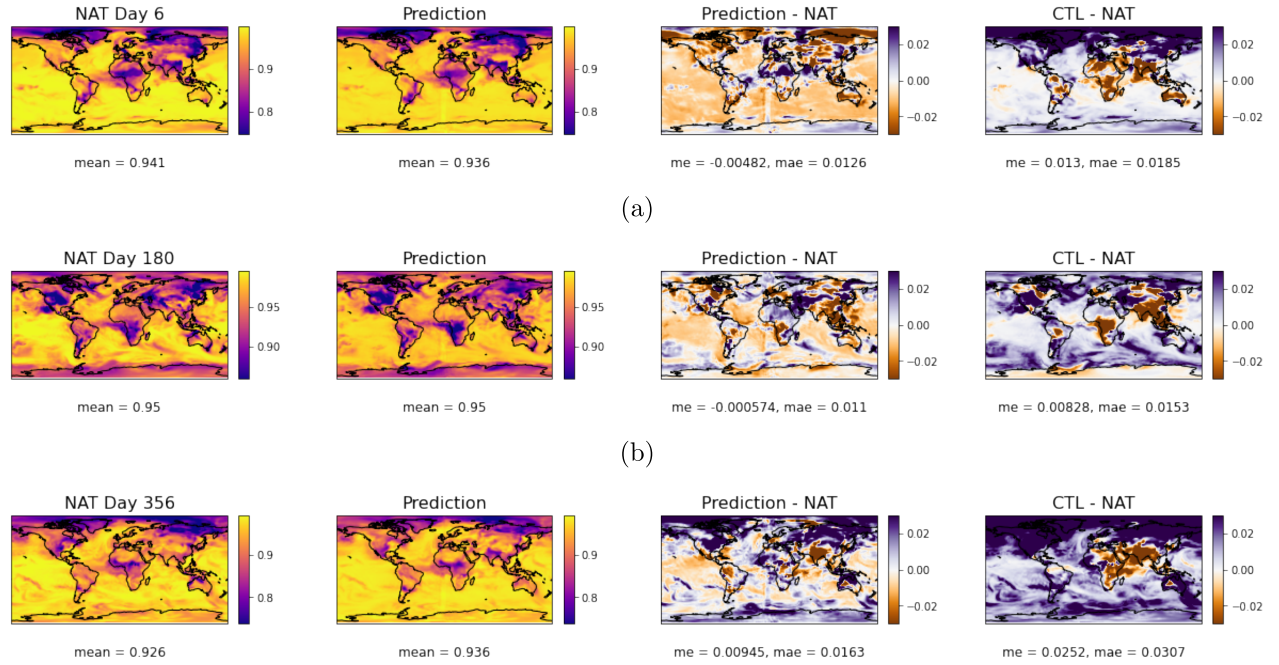
(b)

NAT Day 356 | Prediction | Prediction - NAT | CTL - NAT

mean = 0.112 | mean = 0.112 | me = 0.000246, mae = 0.0206 | me = -0.0152, mae = 0.0372

(c)

Figure 6.2: Global AOD field from NAT and the prediction from the validation set in addition to the differences between the prediction and NAT and CTL and NAT. The fields are from day 6, 180 and 356 of the year and below the fields, the mean is stated. Below the difference maps. the mean error and mean absolute error of the original fields of that day compared to the NAT field are given.

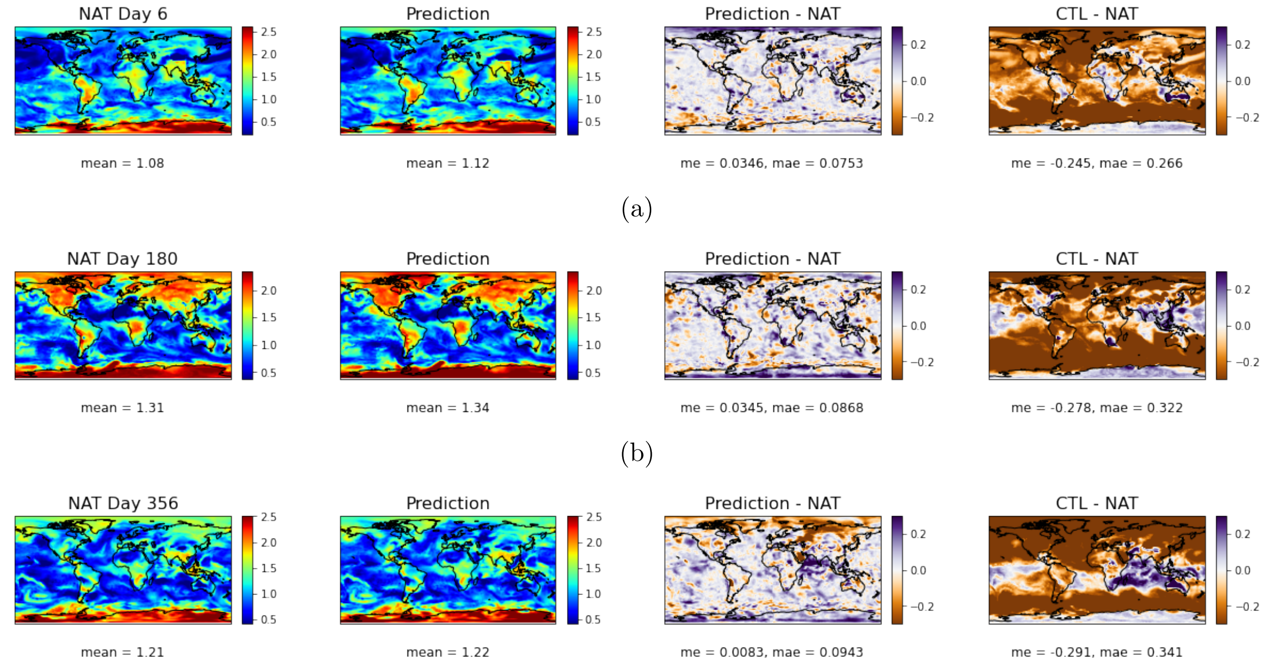


(a)

(b)

(c)

Figure 6.3: Global SSA field from NAT and the prediction from the validation set in addition to the differences between the prediction and NAT and CTL and NAT. The fields are from day 6, 180 and 356 of the year and below the fields, the mean is stated. Below the difference maps. the mean error and mean absolute error of the original fields of that day compared to the NAT field are given.



(a)

(b)

(c)

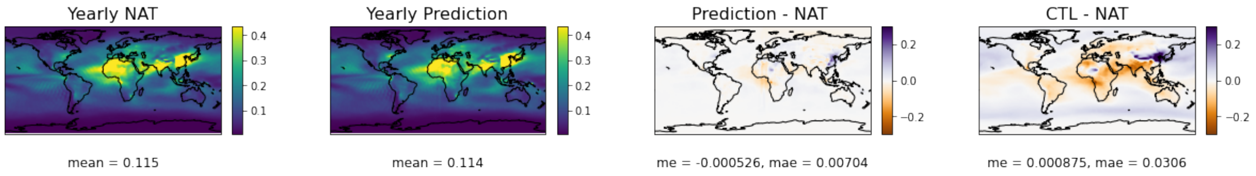
Figure 6.4: Global AE field from NAT and the prediction from the validation set in addition to the differences between the prediction and NAT and CTL and NAT. The fields are from day 6, 180 and 356 of the year and below the fields, the mean is stated. Below the difference maps. the mean error and mean absolute error of the original fields of that day compared to the NAT field are given.

In Figure 6.2, Figure 6.3 and Figure 6.4 the results from the models of AOD, SSA and AE respectively are shown for three samples from the test dataset. Note that for AOD and SSA the NAT field is the original field before the logarithm was taken and the prediction field is the predicted map after taking the exponent. The samples are taken for day 6, day 180 and day 360 from the year, where day 1 is the first sample from the original dataset, before the first and last day were removed. These days were chosen to get an idea on the performance throughout the year. For each sample, we compare the prediction to the truth and also CTL to the truth. The comparison from CTL to truth is included to observe the improvement over our a priori estimate. Inspecting all 9 cases, we notice that the predicted maps are extremely close to the NAT maps, with only some small differences. This is also reflected in how close the global means of the prediction and the truth are to each other. On closer observation of the AOD fields from Figure 6.2, we see that the difference maps are predominately white, which corresponds to a difference close to 0, with a bigger deviation on the areas with a higher AOD. There also seems to be no large bias to either overpredicting or underpredicting. This can also be seen in the very low mean error. Furthermore, the mean absolute error, while slightly higher in day 180 and day 356, is very small. Comparing the MAE relatively to the global mean of NAT, however, all three errors are approximately 18.7% of the mean. If we look at the difference maps for CTL, it is clear it shows larger errors for all three days, also apparent from the higher MAE. Likewise, there is also a bigger bias, although the bias does not seem to be consistent as for day 6, there is a positive bias, while the other two days have a negative bias. Lastly, the CTL fields are closer to the NAT fields for later days, while it is more consistent for the prediction.

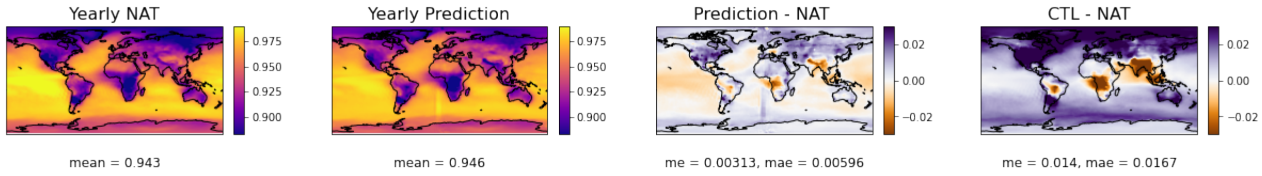
For the results from the SSA model in Figure 6.3, we again find that the trained model has a lower ME and MAE than CTL for all three days. However, unlike with the AOD, there are bigger differences between the predictions and the truths. Comparing the SSA fields with each other, one can see that the model underpredicts in large parts of the globe for the first two days, while it overpredicts in the north on the final day. This is also clear from the global means from day 6 and day 356. Therefore, the model does have a little bias for the first and last day, but almost no bias for day 180. Also the mean absolute errors are not as consistent as they were with the AOD, with a relative MAE of approximately 1.3%, 1.2% and 1.7% for day 6, day 180 and day 356 respectively. While these are lower compared to the relative errors of AOD, they cannot be directly compared to each other, since SSA typically has values between 0.8 and 1.0. Therefore the error range is also smaller. Taking CTL into consideration, we again see that the model performs better than CTL, both in terms of ME and MAE. Furthermore, there seems to be a correlation between the performance of the model and the accuracy of CTL, as the model has higher errors when the CTL field is less accurate. This is not entirely surprising, due to the fact that CTL is an input variable for the neural network.

Lastly, the model for AE again shows good results in Figure 6.4. Once again, the prediction maps and the truth maps look almost identical, with very close global means. The difference maps show that there are no large clusters where the model either overpredicts or underpredicts, except for the north pole and the south pole. Although there does not seem to be a large bias in the maps, the mean errors show that there is a positive bias, especially in day 6 and day 180. This is likely due to the overprediction for the north pole and south pole. However, this is not consistent since the larger error is only present at the north pole on day 6, while it is only present at the south pole on day 180. This might suggest that the model does not have a large bias throughout the year, but that is not fully observable from these three samples. The MAE seems to increase over the year, although this is not necessarily the case, since the MAE's relative to the global mean are 7.0% 6.6% and 7.8% for the start, the middle and the end of the year respectively. Compared to the CTL fields, the model performs a lot better, with the MAE of CTL being almost four times as high, and the CTL run having a large negative bias.

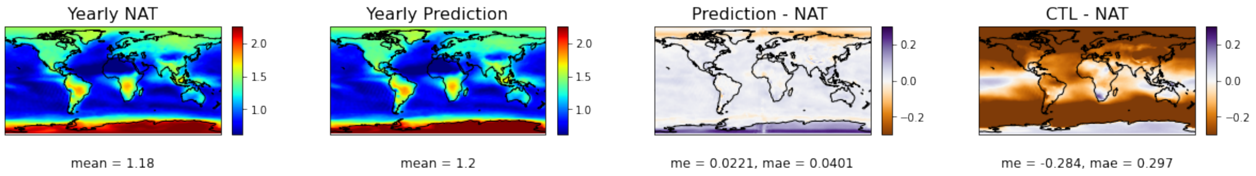
While these results give a first impression on the performance of the model, they do not show the performance for the entire year. Therefore, we let the model get a prediction for the entire validation dataset and took the average global map of those results. This was then also done for both NAT and CTL, which were again compared to one another. These average global maps for the year are shown in Figure 6.5.



(a) AOD



(b) SSA



(c) AE

Figure 6.5: Global field of the yearly average from NAT and the prediction from the validation set in addition to the differences between the prediction and NAT and CTL and NAT. Below the difference maps, the mean error and mean absolute error of the original fields of the yearly average compared to the NAT field are given.

First, we see that the yearly predicted map for the AOD looks almost identical to the yearly NAT map. The yearly maps for the SSA and AE also are very similar to the truth, however the SSA model overpredicts in some northern areas, while the AE model overpredicts near the south pole. For all three models the global means are very close to the global means of NAT. Below the difference maps, the mean error and mean absolute error are stated again. Note that these errors are from the yearly average. Therefore, the underpredictions and overpredictions throughout the year can cancel each other, which leads to a lower error. The difference map for the AOD shows that the difference is close to 0 for almost the entire globe, except for some small areas. Also the CTL map has very low errors, although the overall errors are a little higher and there are bigger areas with a higher error. The low errors for the prediction also further reinforce the hypothesis that the model has low bias. Also the MAE is lower than the MAE's seen in the day predictions, likely a byproduct of the low bias and taking the average of the year. The year MAE for CTL is also lower than the day MAE's, but the difference is not as big. This is another byproduct of the higher bias of CTL. Since CTL has lower AOD values for most samples, the yearly average will also have lower values. In contrast, the model, which is more balanced with regards to underprediction and overprediction, will have a yearly average closer to the truth, since an overprediction over an area in one day can cancel out an underprediction over the same area in another day. Consequently, the MAE of the yearly average for the prediction decreases by a larger amount compared to the MAE of the yearly average for CTL.

The yearly difference map for SSA again seems to have overall lower errors than the daily maps, again implying a lower day to day bias. Like we saw in the daily maps for the SSA values, there is some overestimation on northern areas. This overestimation, however, is greatly reduced compared to the overestimation of the CTL run. Additionally, the CTL field has an overall higher error in the other regions. There is a clear area at the south pole where CTL is closer to NAT compared to the rest of the CTL field. Lastly, the maps show a clear line around longitude 0 where the model overpredicts. This is caused by the orientation of the original data fields on which the model was trained, where longitude 0 was at the left boundary of the maps. Since these maps are shifted such that longitude 0 appears in the middle, a slight discontinuity is likely the cause of this line.

Finally, the difference map for AE clearly shows underprediction and overprediction of the model at the north pole and south pole respectively. This proves that the model has a positive bias for the south pole, which was not entirely apparent from the daily fields. The negative bias for the north pole is not as big, which can also be observed from the ME. This area is also visible in the model prediction. Interestingly, CTL has a lower error at the south pole than the prediction. A possible explanation for this phenomenon is the distribution of the SPEXone measurements. As could be seen from the SPEXone samples in Chapter 4, the southern and northern parts of the map are barely covered by SPEXone. As a result, the model only has the CTL field as a reference for those areas. While CTL clearly performs better at the south pole, this does not necessarily imply a better performance of the model in this region too. Since CTL for the network input is scaled to the interval $[0, 1]$, the network is mainly able to capture the structure of CTL and not the actual values. Therefore, the maximum value of the field is not directly known, as the model has SPEXone measurements in the area for reference. Again the MAE for the predicted map is lower compared to the daily results. Apart from the north and south pole, the model performs really well, as there is only a very small error in the other regions. The AE model has the biggest improvement from the CTL run, with an MAE of less than 14% of the CTL MAE. Just like the daily CTL AE fields showed, the CTL field is lower than the NAT field for the most part of the globe.

Since taking the yearly average lowers the overall errors across the map, it can give a skewed view of the performance. To get a better understanding on the performance on the entire test set, we also calculate the MAE across the entire year, instead of only for the yearly average map. This is done as follows:

$$\mathrm{MAE}_c = \frac{1}{n}\sum_{i=1}^{n} |P_i - \mathrm{NAT}_i|, \tag{6.1}$$

where MAE_c is the MAE across the year, n is the number of samples in the validation set and P_i and NAT_i are the prediction field and NAT field respectively. These maps are given in Figure 6.6 for both the predictions and CTL.

(a) AOD

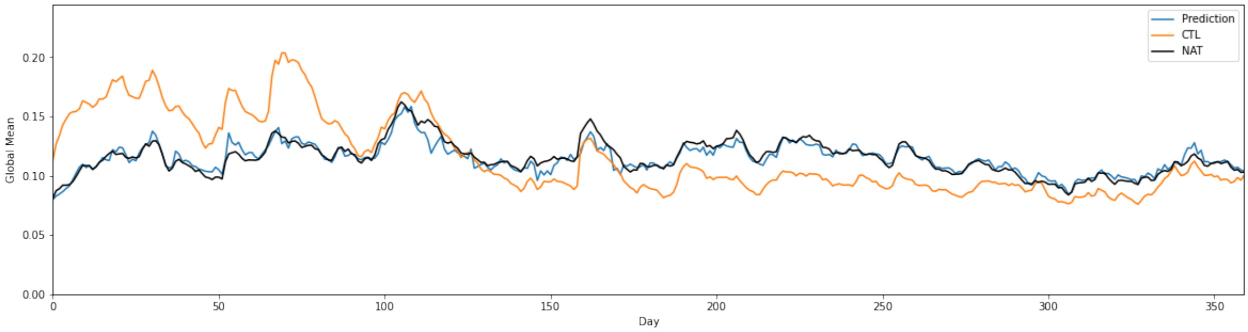
(b) SSA

(c) AE

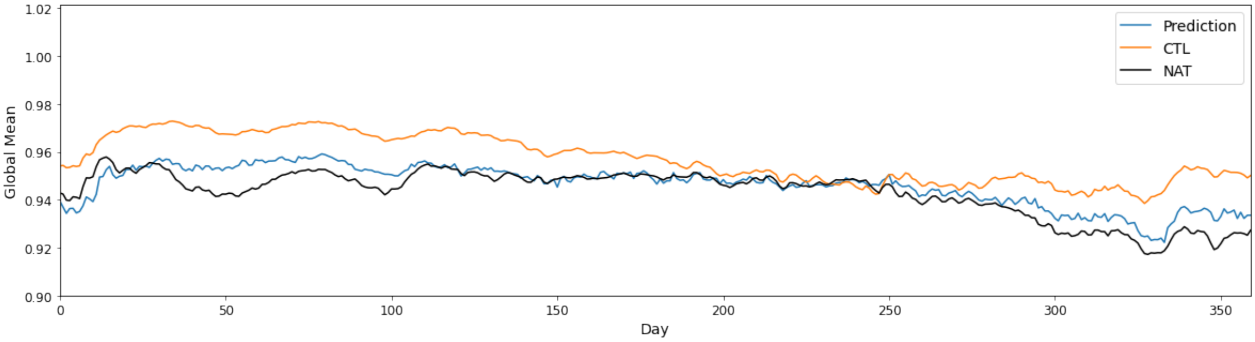
Figure 6.6: Cumulative MAE of the models predictions over the validation set and CTL for AOD, SSA and AE. Below each map, the global mean of the cumulative MAE is stated.

Figure 6.6 gives a good overview on where the model has more trouble making a correct prediction. The figure shows that AOD performs the best among the three models, with overall a low MAE_c, with one small area in East Asia where the error is larger. This area was also visible in the yearly map, although not as pronounced due to it being a yearly average. The presence of the higher error in this area is likely due to the area having the maximum value in most days. As a result, we get a similar scenario as in the yearly average AE fields at the south pole, although, the scale is smaller. This in combination with the poorer performance of the input of CTL over this area, is likely the reason for the higher error. The mean of the cumulative MAE is also less than half of the mean of the cumulative MAE of CTL. In the CTL map, the area with a higher error is also visible, however it is a little larger. Furthermore, the overall error across the field is slightly higher. The error field for SSA has relatively low errors across the oceans, however over land, the errors are higher. This is again likely caused by the poorer performance of CTL over these areas, which is then also translated to a worse performance in the model, due to CTL being an input to the model. Compared to CTL it does not perform as well as AOD, with the mean being more than half of the mean for CTL. Nevertheless, it is still a good improvement over the control run. The figure for AE again shows the shortcomings of the model with regard to the south pole region, where it is clearly higher than the other parts of the world. Also over the north pole, the error is slightly higher. These results correspond with what we saw in Figure 6.5. Likewise, we again see the major improvement over CTL, which only has a better performance over the south pole. For all three cases, we once again see the lower bias for the model compared to CTL, since the increase from the MAE of the yearly average to the MAE_c is higher for the model than it is for CTL. This again is caused by the underpredictions cancelling out the overpredictions in a higher degree for the model.

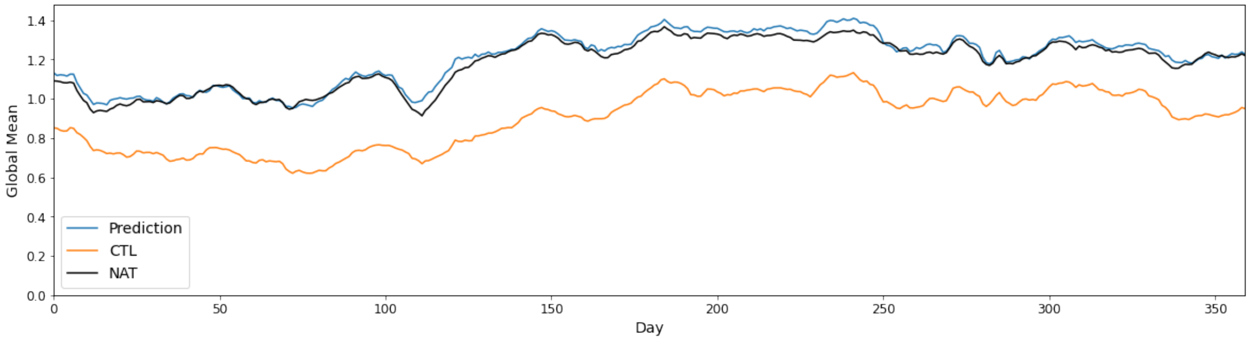
Next we plot two time series for the full validation set. Firstly, we plot a time series of the global mean for the prediction, NAT and CTL in Figure 6.7. The second plotted time series is one for the correlation between prediction and NAT and between CTL and NAT in Figure 6.8. For all AOD, SSA and AE, the time series for the predictions are closer to the time series for NAT than the CTL time series are. Especially the series for AOD and AE are really close. The series for SSA however does have some areas where it deviates a little, namely around the start of the year and near the end, which could also be seen in the differences in global means in Figure 6.3, where the model has a better performance on day 180 than on day 8 and day 356. As discussed previously, the higher errors in these days is likely due to the higher errors of CTL in those days as well. It is also notable that the CTL means for AOD and SSA are quite close to the NAT means in some intervals, while the CTL means for AE consistently have approximately the same difference. The CTL series for AE also seems to have a more similar pattern to the NAT series compared to the series for AOD and SSA. This similarity can also be observed from Figure 6.8, where we can see that CTL for AE has on average a higher correlation than CTL for AOD and SSA. The figure also shows that the correlation of the prediction is always higher than the correlation of CTL for AE, while it is higher for almost all days for AOD and SSA. Furthermore, the days in Figure 6.7 where the global mean of the prediction and CTL are almost the same, around day 100 for AOD and around day 240 for SSA, are also present in the correlation time series in the form of small differences in correlation. In Table 6.1, the average correlation over the year is shown for both the model predictions and CTL. These again show that the results from the model are higher correlated for all three cases. It is also important to note that the model seems to perform better with a higher correlation of CTL. This is likely due to the fact that CTL is used as an input for the network, which means that a better CTL field also results in a better prediction.



(a) AOD

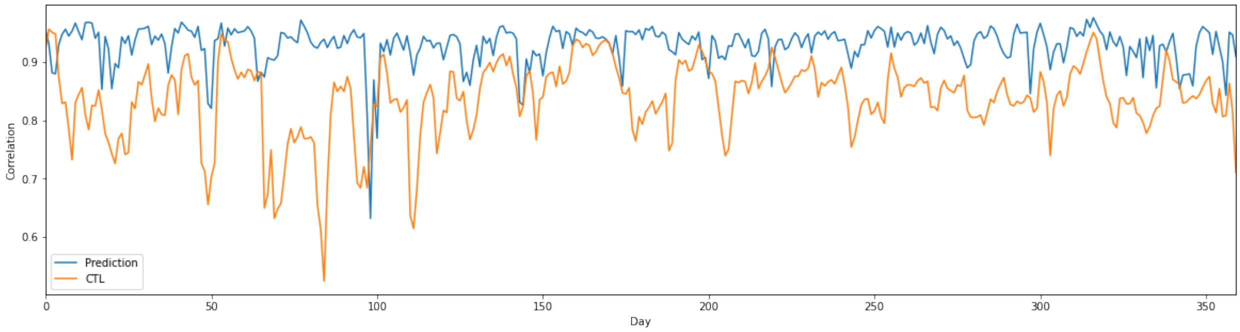


(b) SSA

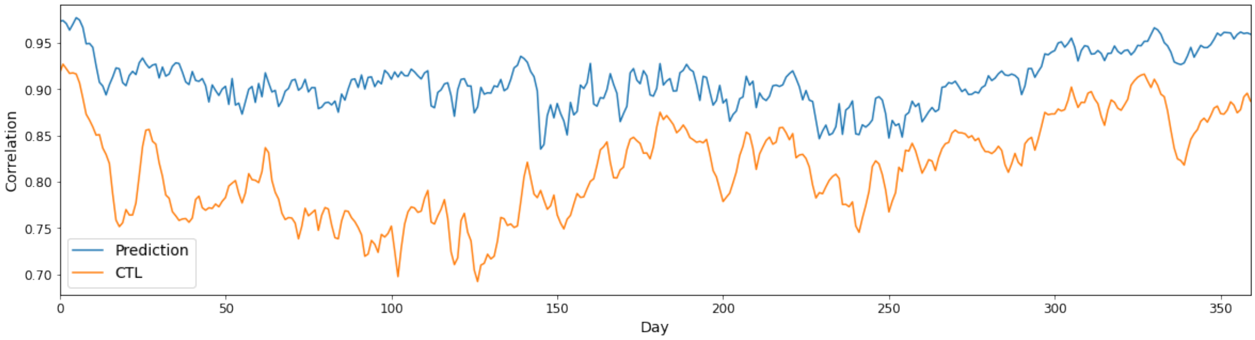


(c) AE

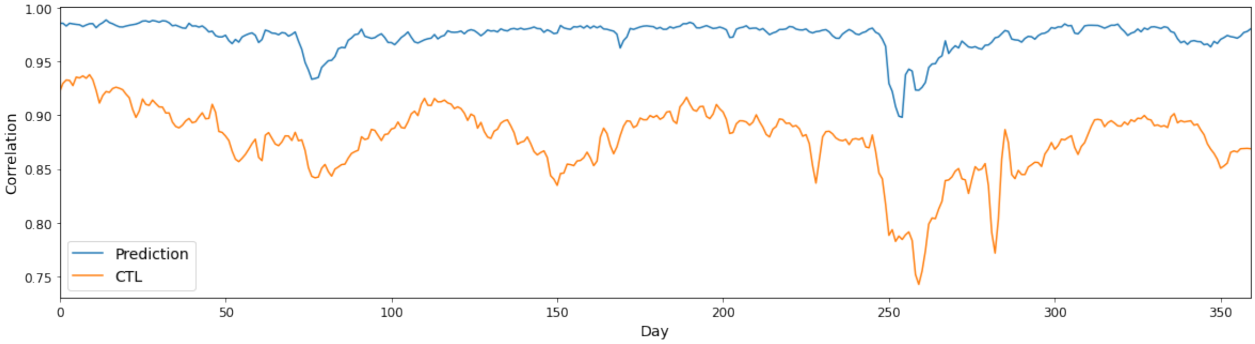
Figure 6.7: Time series of the global mean for the prediction, NAT and CTL for AOD, SSA and AE.



(a) AOD



(b) SSA



(c) AE

Figure 6.8: The time series of the correlations between prediction and NAT and between CTL and NAT for AOD, SSA and AE.

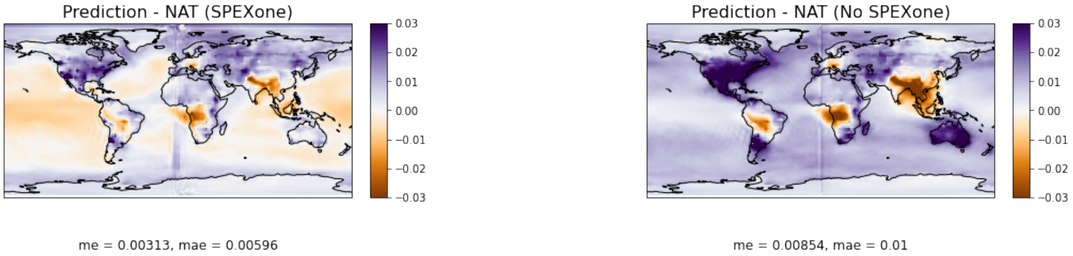
	AOD	SSA	AE
Prediction	0.929	0.909	0.974
CTL	0.837	0.814	0.878

Table 6.1: The average of the correlation time series from Figure 6.8 for both the prediction and CTL.

6.3 Influence of SPEXone

As we saw in the results for the model, there seems to be a correlation between the performance of the model and the accuracy of CTL. Therefore, it might also be the case that the network is able to improve CTL, without the input of SPEXone. To test whether SPEXone improves the results and by how much, we also trained the model using only the CTL fields as input. The network had almost the same architecture as in Figure 5.1, however it only had one encoder part, since we only have one input. In Figure 6.9 the difference maps of the yearly mean are shown for both the previous model, which had both CTL and SPEXone as inputs, and for the model trained without SPEXone as input. Figure 6.10 shows the cumulative MAE over the year for both models. Lastly, in Table 6.2, we state the average correlation over the year for the models with and without SPEXone. From both the figures and from the results in the table, it is clear that the model without SPEXone as input, performs significantly worse. For AOD, SSA and AE the MAE of the yearly mean map is at least almost twice as high. The difference in the cumulative MAE is not as high, however there still is a noticeable increase compared to the original model. We do note that the trained model with only CTL as input still has lower errors and higher correlation than CTL itself for both SSA and AE. This is not the case for AOD, where the model seems to perform significantly worse with errors being more than 10 times higher and the correlation being almost similar. The reason for this can be seen in Figure 6.9, where we notice a small area on the right side of the map where the model suddenly overpredicts, while it underpredicts in the areas around that part. Upon further inspection, it was found out that this is due to one day in the testing data, where it reached a value of over $1 \cdot 10^6$ in that area, while the model behaved normally in the rest of the globe and for the other days. This was likely caused by an error or a bug caused during training, as there is no other reasonable explanation. Removing this day from the dataset resulted into fields more similar to the CTL fields, although it was still not a significant improvement over CTL. Without that sample, the model had a ME of -0.0288, a MAE of 0.0303 and a mean of the cumulative MAE of 0.047. The better performance of the model with SPEXone is expected, as SPEXone contains actual values of the NAT fields, which can form a basis for the range of the present values. These figures again show a clear line at longitude 0 for both models, which is due to the initial orientation of the data fields as explained in the previous section.

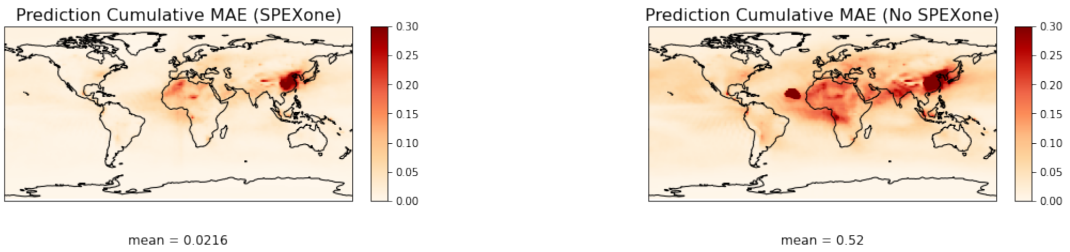
(a) AOD



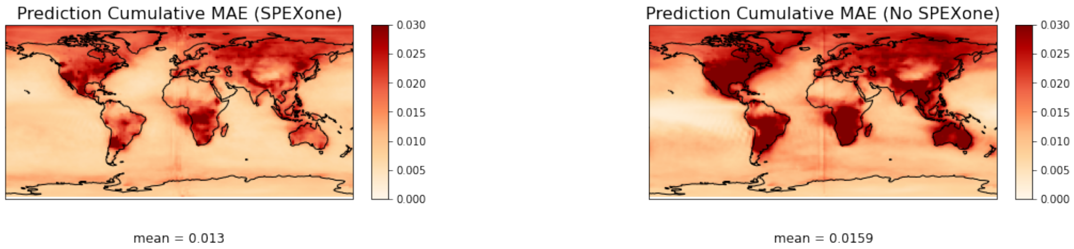
(b) SSA

(c) AE

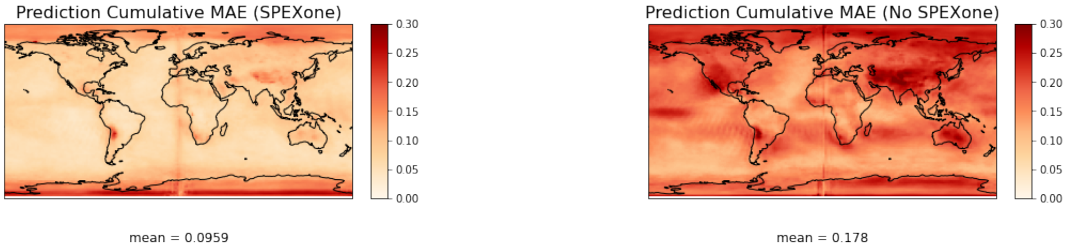
Figure 6.9: Training loss and validation loss against the epochs for the models for AOD, SSA and AE. The loss function used is MAE.



(a) AOD



(b) SSA



(c) AE

Figure 6.10: Training loss and validation loss against the epochs for the models for AOD, SSA and AE. The loss function used is MAE.

	AOD	SSA	AE
With SPEXone	0.929	0.909	0.974
Without SPEXone	0.835	0.885	0.930

Table 6.2: The average correlation over the year for both the model with SPEXone as input and the model without SPEXone as input.

6.4 Results from other Experiments

The aforementioned results are considered to be the results from the best model. In this section we also give a brief overview of two other performed experiments. The first experiment, denoted as Exp001 was almost identical to the final experiment. The model architecture is the same and the only difference is in the training data. The training data for Exp001 consisted only of the original SPEXone, CTL and NAT fields and did not also include the flipped images. The second experiment, denoted as Exp002, again had the same model architecture. The training dataset again only had the original images, and no flipped fields. Additionally, the NAT fields of the training data were different. The NAT fields for Exp002 were from one of 32 ensemble members used in data assimilation for the original NAT model. This implies that the NAT fields used in the training data are more related to the NAT fields in the testing data. Another minor difference in both experiments is that the logarithm was not taken of the NAT fields for SSA. The yearly average ME, yearly average MAE, cumulative MAE mean and average correlation are given for the CTL, final model, discussed previously, that model trained without SPEXone, Exp001 and Exp002 in Table 6.3, Table 6.4 and Table 6.5 for AOD, SSA and AE respectively. Note that in Table 6.3 the stated errors of the model without SPEXone are taken over the test dataset from which the day with the enormous outlier was removed.

	Yearly Average ME	Yearly Average MAE	Cumulative MAE mean	Average Correlation
CTL	0.000 875	0.0306	0.0515	0.837
Final model	−0.000 526	0.007 04	0.0216	0.929
Without SPEXone	−0.0288	0.0303	0.0470	0.835
Exp001	−0.003 73	0.009 73	0.0272	0.894
Exp002	0.000 193	0.005 55	0.0228	0.938

Table 6.3: The AOD errors and correlation for CTL, the final model, the model trained on only CTL and the experiments Exp001 and Exp002. The errors for the model without SPEXone are taken over the dataset from which the day with the tremendous error was omitted.

	Yearly Average ME	Yearly Average MAE	Cumulative MAE mean	Average Correlation
CTL	0.0140	0.0167	0.0210	0.814
Final model	0.003 13	0.005 96	0.0130	0.909
Without SPEXone	0.008 54	0.0100	0.0159	0.885
Exp001	0.004 72	0.007 16	0.0144	0.894
Exp002	−0.000 423	0.002 36	0.009 58	0.938

Table 6.4: The SSA errors and correlation for CTL, the final model, the model trained on only CTL and the experiments Exp001 and Exp002.

	Yearly Average ME	Yearly Average MAE	Cumulative MAE mean	Average Correlation
CTL	−0.284	0.297	0.338	0.878
Final model	0.0221	0.0401	0.0959	0.974
Without SPEXone	0.001 12	0.0975	0.178	0.930
Exp001	0.006 72	0.0525	0.121	0.961
Exp002	0.002 82	0.0319	0.0789	0.985

Table 6.5: The AE errors and correlation for CTL, the final model, the model trained on only CTL and the experiments Exp001 and Exp002.

The tables give a good perspective on the performance of Exp001 and Exp002 compared to the final model. In all three cases Exp001 performs worse than the final model. This is due to the network being trained on fewer data as well as less varied data. This confirms that the flipped images actually increase the performance of the model. For AOD and AE, the increase is quite noticeable, while the increase for SSA is relatively small. As for Exp002, the errors are lower, while the correlation is higher for all three cases except for the cumulative MAE mean for AOD. This is expected, due to the similarity of the training output and the testing output. This was also the reason for not opting to choose this model over the final model, since the ensemble member of NAT and NAT itself are too similar to each other. Therefore, the validation is not entirely fair. Since the results of the final model are still good and not that much worse than those of Exp002, especially for AOD, the final model is a better indication of the performance of the model.

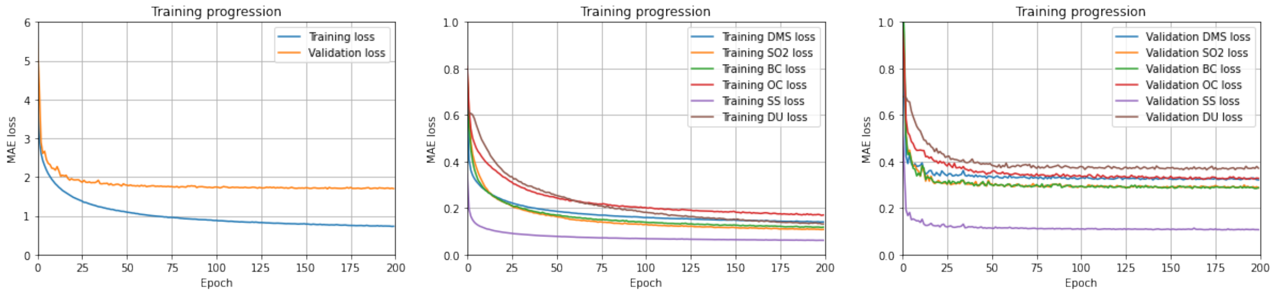
Chapter 7

Results for the Emissions Problem

This chapter follows a similar structure to the previous chapter. First the training progression and results from the model are discussed in Section 7.1 and Section 7.2 respectively. Lastly, results from other performed experiments are presented in Section 7.3.

7.1 Training Progression

The model was trained on a GPU cluster of the Delft High Perfomance Computing Centre (DHPC) [19]. On the GPU node, two separate GPU's were simultaneously used to reduce computation time. The model ran for 200 epochs with a batch size of 16. Furthermore, the mean absolute error was used for the loss function. The Adam optimizer for Keras was used with a learning rate of $1 \cdot 10^{-3}$ and a decay of $\frac{1 \cdot 10^{-3}}{200}$. The training progression of the model is shown in Figure 7.1a. The training loss and validation loss in this figure are the sums of respectively the training and validation losses of the 6 different outputs. In Figure 7.1b the training losses for each different output is shown and Figure 7.1c the validation loss for each output is shown. After the final epoch, the validation loss is equal to 1.70, while the training loss has a value of 0.73. The validation loss also barely changes in the final epochs, implying that the model converged. The losses for each species also show almost identical values for the training and validation losses. Furthermore, it can be observed that OC is the highest contributor to the total loss, while DU is the highest contributor for the validation loss. For both the training and validation loss, SS contributes the least. Note that this does not imply a better performance for SS compared to OC or DU, since it could also be a result of higher OC or DU values and lower SS values. To get a better understanding of the performance of the model for each species, we again test the model using the validation dataset.

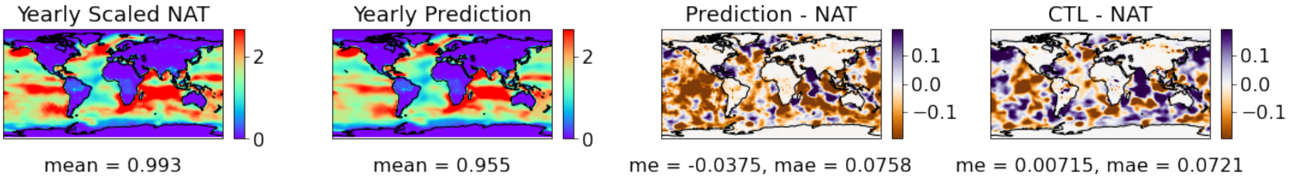


(a) Full Training and Validation loss (b) Training loss for each species (c) Validation loss for each species

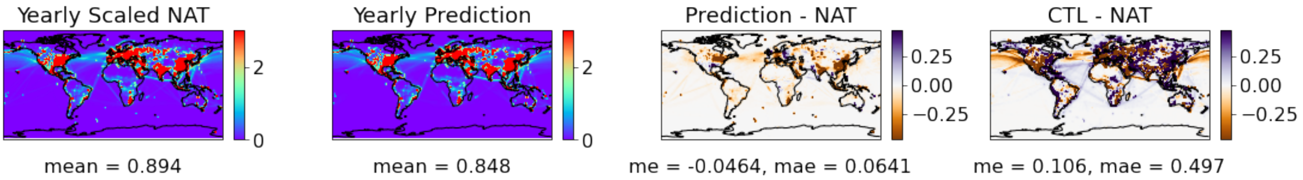
Figure 7.1: Training loss and validation loss against the epochs of the model, the training loss for each emission species and the validation loss for each emission species. The loss function used is MAE.

7.2 Model Predictions

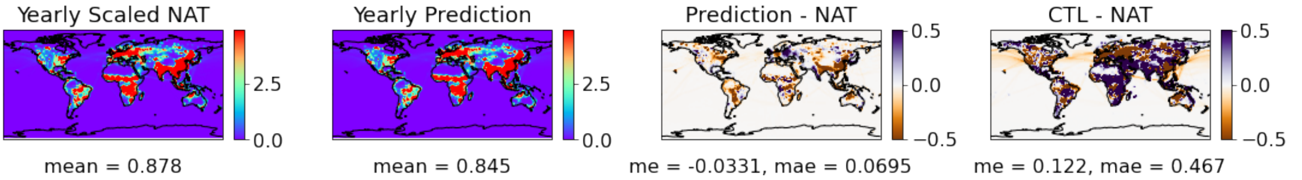
Due to a higher amount of outputs compared to the aerosol properties problem, we omit the "daily" results and instead start with the yearly average maps of the entire validation set. In Figure 7.2, the yearly average maps of NAT and the model prediction are shown in addition to the differences of the yearly predicted maps and the yearly NAT maps and the differences of the yearly CTL maps and the yearly NAT maps. Below the yearly average maps, the global means are stated and below the difference maps, the mean error and the mean absolute error are stated.



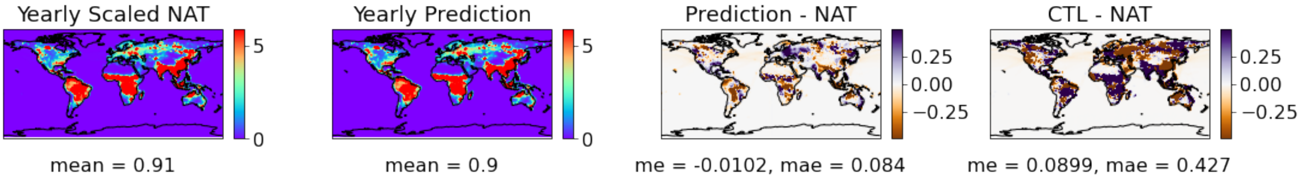
(a) DMS



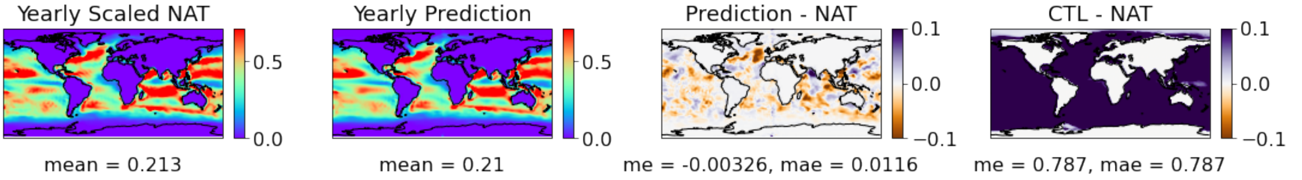
(b) SO_2



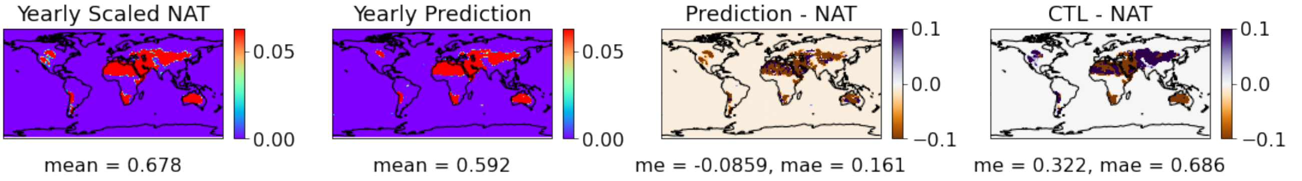
(c) BC



(d) OC



(e) SS



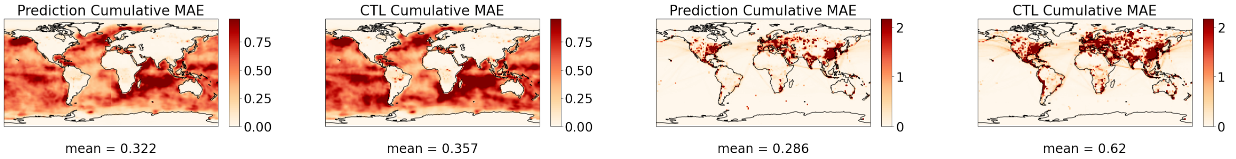
(f) DU

Figure 7.2: Global field of the yearly average from NAT and the prediction from the validation set in addition to the differences between the prediction and NAT and CTL and NAT. Below the difference maps, the mean error and mean absolute error of the original fields of the yearly average compared to the NAT field are given.

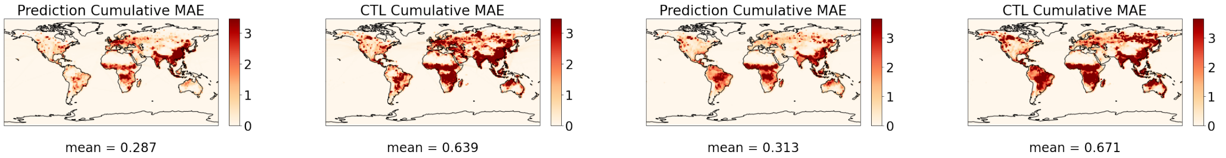
A first inspection of the figure shows that the global maps of the model prediction look very similar to the global maps of NAT, with some small differences. This implies that the model is able to predict where the emissions are present, however the exact value of the emissions can still differ due to the chosen ranges for the maps. For example, the DU emissions look almost the same, however the global mean shows that the model underpredicts for this species. This is difficult to conclude from the maps themselves, since there are no DU emissions on most parts of the globe. Therefore this difference is likely only coming from the visible regions with DU emissions. The difference map for DU, where we can see that, while the model is able to predict where these emissions are present but not the exact amount, confirms this. The difference map of CTL for DU, shows the same patterns as those in the difference map of the model. The MAE of the CTL map is more than 5 times as high as the MAE for the prediction, indicating a improvement of the model. The ME indicates that CTL has approximately the same bias as the model although the model is more negatively biased while CTL has a more positive bias.

On closer inspection of the difference maps for the other emission species, we observe very low differences for SO_2, BC, OC and SS, while a more noticeable difference for DMS. SO_2. BC, OC and SS all have a MAE of less than 10% of their respective global mean, with the MAE of SS being almost 5%. Both SO_2 and BC have a noticeable negative bias. OC and SS also have a negative bias, although it is smaller than the bias for SO_2 and BC. Also for these four emissions species does the model have a clear improvement over CTL. This is especially the case for SS, where CTL is very dissimilar from the truth. For SO_2, BC and OC the model performs noticeably better over North America, Asia and Europe, having smaller and fewer regions with larger errors. For SO_2 and BC the model also makes a smaller error over the Pacific Ocean and the Atlantic Ocean, where we observe some areas where CTL has a lower or higher value.

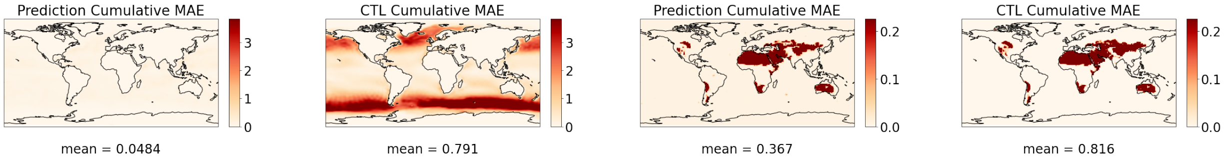
The model does not have a better performance for DMS compared to CTL. While the MAE is still less than 10% of the global mean, the prediction has both a higher bias and a slightly higher MAE than CTL. Therefore, the model might still have a good performance, however the CTL emissions are already very accurate, which means it is difficult for the network to increase the results over CTL. Furthermore, DMS is a precursor gas responsible for about 3% of sulfate (SO_4) in the atmosphere and SO_4 AOD contributes only approximately 10-20% to the global AOD. Therefore, DMS plays only a small role in the global field, which may be another reason for the lower performance of the network compared to CTL. These maps also show the reason for the low contribution of SS to the total loss and the high contribution of OC and DU, as SS has the lowest values, while OC and DU have higher values among the species. Next we calculate the cumulative MAE across the full test dataset again. These results are displayed in Figure 7.3.



(a) DMS (b) SO_2



(c) BC (d) OC

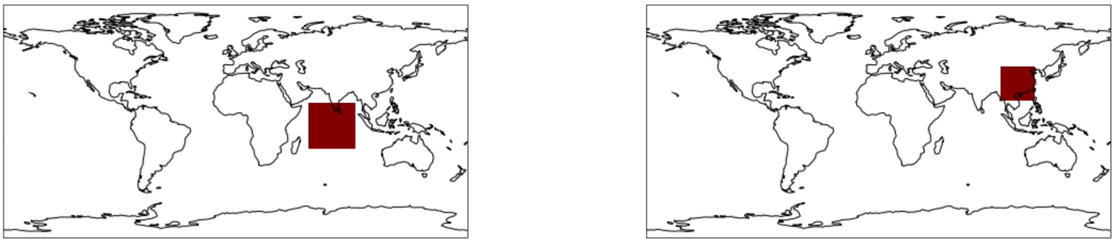


(e) SS (f) DU

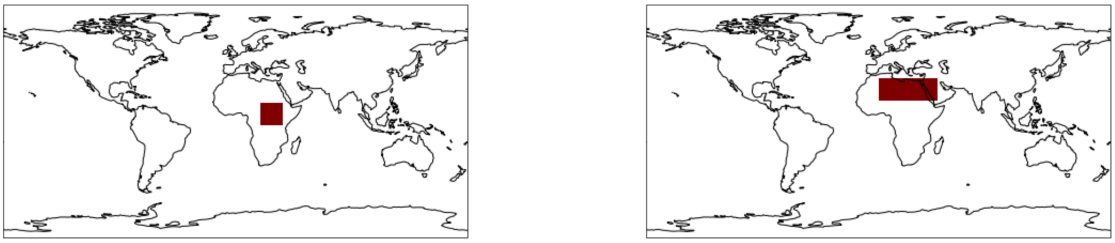
Figure 7.3: Cumulative MAE for the model prediction over the test dataset and CTL for DMS, SO_2, BC, OC, SS and DU. Below each map, the global mean of the cumulative MAE is stated.

The first thing we notice from Figure 7.3, is that the model has a lower cumulative MAE for DMS than CTL has. Since the MAE of the yearly average map for the model prediction was higher than the MAE for CTL, this implies less values for the prediction cancel each other out. Thus, the model does have a higher bias than CTL, which we already observed in Figure 7.2. The model has a lower cumulative MAE in the Pacific Ocean and in the southern parts of the Indian Ocean. The model also has slightly lower errors in the Atlantic Ocean, although the difference is not as apparent. The model has an improvement over CTL for the other species. Again SS has the biggest improvement, with a cumulative MAE mean of more than 15 times smaller than the mean of CTL, while for the other 4 species the CTL mean is a little more than twice as high. The cumulative MAE of CTL for SO_2, BC and OC is noticeable in the same regions as the cumulative MAE of the model, although it is higher and more pronounced. From these figures we can also conclude that the model has a higher bias than CTL, since the increase from the MAE in Figure 7.2 to the cumulative MAE mean is smaller for the model compared to CTL.

To get a better understanding of the model performance for each individual day of the validation data, we plot a time series of NAT, CTL and the prediction for each species. This time series will be of the mean emissions in a certain area depending on the species. Since for all species there are large parts of the globe where no or very little emissions are present, this approach was chosen instead of using the global mean. The chosen area for each species is an area where that particular species has high emissions. This is the Indian Ocean for DMS and SS, East Asia for SO_2, Central Africa for BC and OC and the Sahara region for DU. The defined regions are shown in Figure 7.4. Furthermore, we again plot a time series of the correlation between the model prediction and NAT and between CTL and NAT. Note that these correlations are taken over the full map and not over the regions specified earlier. The time series for the regional means are given in Figure 7.5 and the time series for the correlations are shown in Figure 7.6.

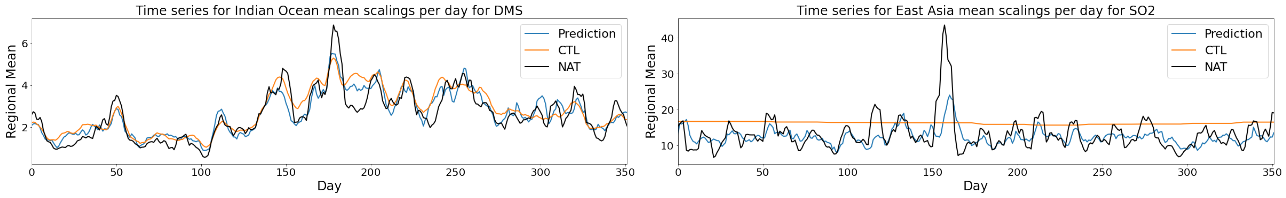


(a) Defined Indean Ocean Region (b) Defined East Asian Region

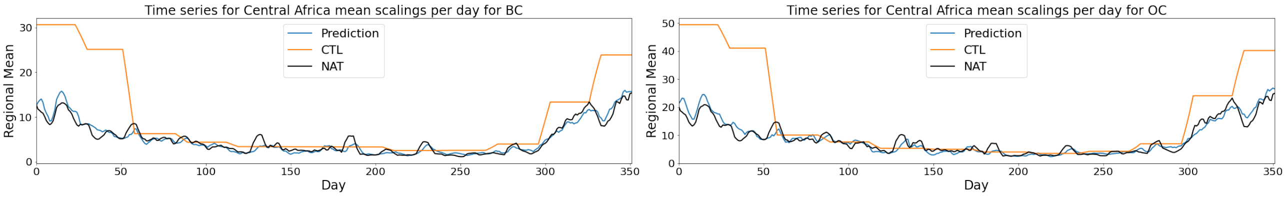


(c) Defined Central Africa Region (d) Defined Sahara Region

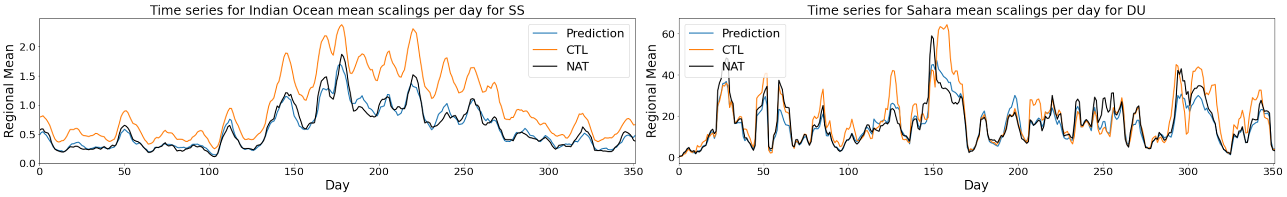
Figure 7.4: The defined regions used to calculate the regional means over for the time series.



(a) DMS (b) SO_2

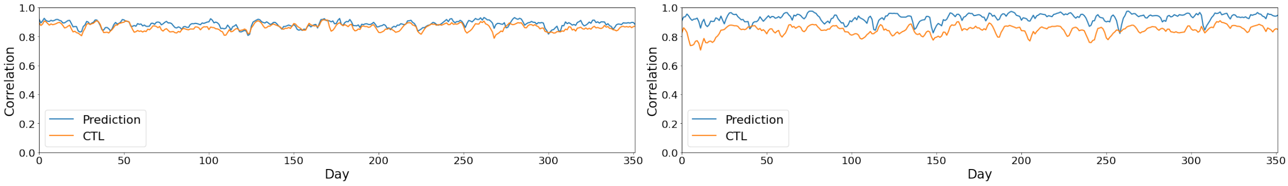


(c) BC (d) OC

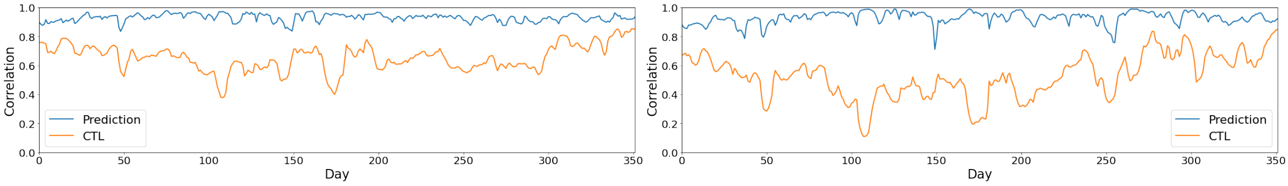


(e) SS (f) DU

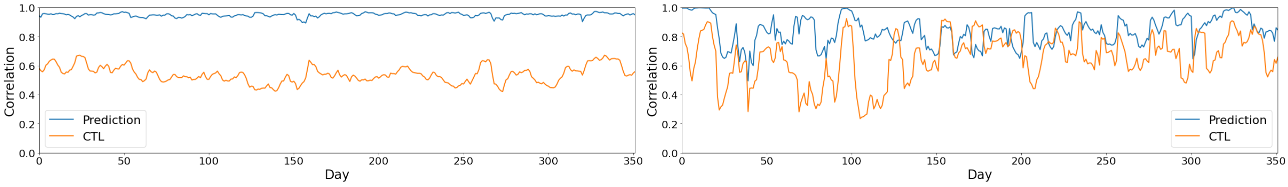
Figure 7.5: Time series of the regional mean for the region stated above each time series. The time series for NAT, CTL and the model prediction are given for each species.



(a) DMS (b) SO_2



(c) BC (d) OC



(e) SS (f) DU

Figure 7.6: Time series of the correlations between prediction and NAT and between CTL and NAT for each emission species.

The time series in Figure 7.5 show that overall the regional means of the prediction have the same behaviour as NAT, while CTL varies more depending on the species. The CTL time series for SO_2, BC and OC are vastly different from NAT, while the series for DMS, SS and DU have similar patterns. Note that while the CTL time series for SO_2, BC and OC have large intervals where the means are constant, this does not mean that the emissions in the areas are also constant during these periods. Since all emissions are scaled with the mean CTL emissions, including the CTL emissions themselves, this could simply mean that an increase or decrease over the areas comes paired with a global increase or decrease. As the large differences between the cumulative MAE of the predicted DMS and the CTL DMS were primarily over the Pacific Ocean and the south of the Indian Ocean, these differences are not present in the time series. Therefore, both the model time series and the CTL time series are very close to the truth. However, the series still shows the reason for the higher MAE_c of CTL, since the time series of the prediction looks slightly closer to the NAT time series. The model also has some apparent inaccuracies for SO_2, in particular around day 120 and 160, the model greatly underpredicts.

Both BC and OC show a similar pattern, with higher values at the start and at the end of the year. This is likely due to a large scale of savannah and forest fires between December and February due to dry monsoon periods. For both species, the model time series is very similar to the truth, with only some small dissimilarities. Also the series for SS is very similar to the truth. The CTL time series also shows that CTL does still have a similar pattern over the Indian Ocean, despite the high errors. The series also nicely shows that CTL does indeed consistently have higher values than NAT, which we could also observe in Figure 7.2. DU again shows similar values between the predictions, CTL and NAT. Both CTL and the predictions are close to the truth, although the model predictions look slightly more accurate, similar to DMS. The correlation time series in Figure 7.6 show a clear improvement of the model over CTL for all species except DMS. It is worth noting that, while the model does not have better correlations for DMS, it is partially due to the already high correlations of CTL, with an average of 0.862. The other species have consistently higher correlation with less and smaller fluctuations, with the exception of DU. The average correlation of the model and CTL are stated for each species in Table 7.1. The table shows that the model does have a slightly better correlation on average for DMS. The average correlation for the other species is very high, with little to no reasonable room for improvement, with the exception of DU, which has the lowest correlation among the species.

	DMS	SO_2	BC	OC	SS	DU
Prediction	0.885	0.930	0.930	0.929	0.950	0.840
CTL	0.862	0.841	0.653	0.523	0.540	0.652

Table 7.1: The average of the correlation time series from Figure 7.6 for both the prediction and CTL.

7.3 Performing Different Experiments

Similar to Section 6.4, we again perform and evaluate two additional experiments. The same network was trained for both experiments, although they were trained on different data. The first experiment, denoted as Exp001, was trained using a data setup similar to the aerosol properties problem. Instead of using the first trained model to obtain global AOD, SSA and AE fields for training, this network was trained directly using the CTL fields, as well as their gradients, and the SPEXone measurements. Since the emissions problem uses average emissions of 7 days, the CTL fields were also averaged. Likewise, the SPEXone measurements were averaged over 7 days. Due to the fact that the majority of the globe is not covered by SPEXone in one day and therefore it is unlikely for SPEXone to cover the same position more than once in 7 days, the 7 day average of SPEXone measurements often results in a field with less missing values. A sample of 7 day average SPEXone measurements for AOD, SSA and AE is shown in Figure 7.7. These averaged SPEXone measurements were concatenated along the channel axis to CTL and their gradients for AOD, SSA and AE respectively. Therefore, this experiment had 4 channels as input for each input instead of 3. This experiment was performed to investigate how much can be gained from a full global field for AOD, SSA and AE, instead of sparsely sampled field, with values closer to the truth than CTL.

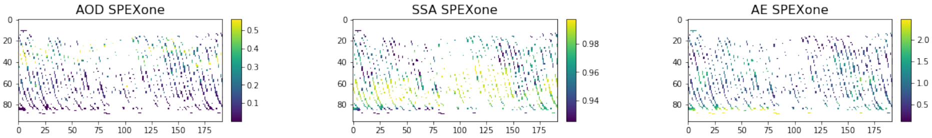
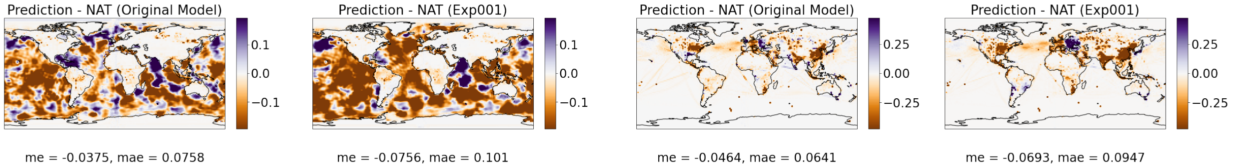


Figure 7.7: The average SPEXone measurements over 7 days for AOD, SSA and AE.

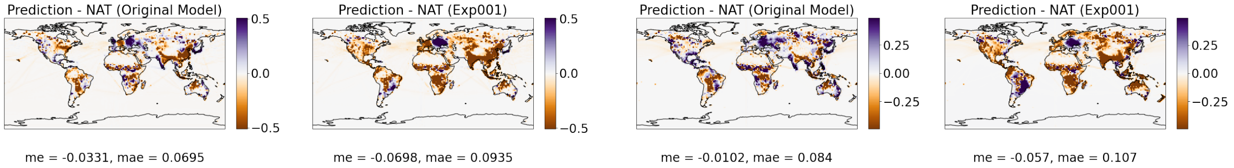
The second experiment, denoted as noSPEX, again used the CTL fields and their gradients, but SPEXone was omitted to investigate the importance of sparsely sampled SPEXone in addition to only CTL fields. This approach was chosen to test how well the model is able to perform without any SPEXone measurements, and only fields which do are not necessarily related to the emissions. Note that the CTL fields do not differ depending on the ensemble member and only on the sample in the ensemble member. Therefore, multiple outputs correspond to one input, including the output from the validation set.

7.3.1 Results from Exp001

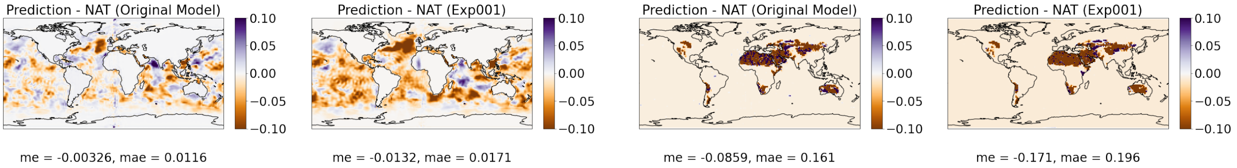
First we evaluate the performance of the Exp001 model on the validation set. The yearly average differences are shown in Figure 7.8 for both the original model and Exp001. Figure 7.10 shows the regional time series for Exp001.



(a) DMS (b) SO_2

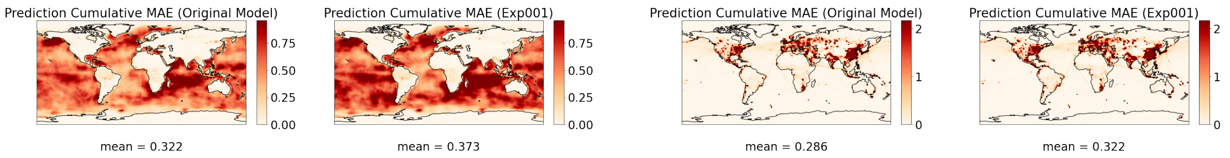


(c) BC (d) OC

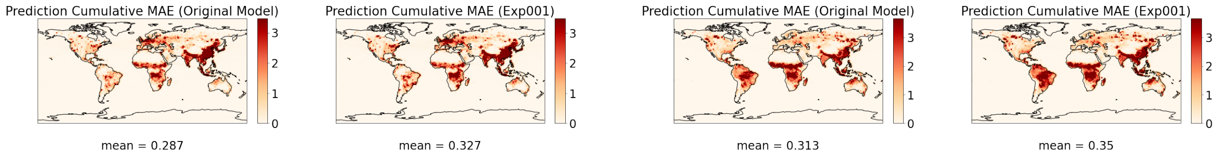


(e) SS (f) DU

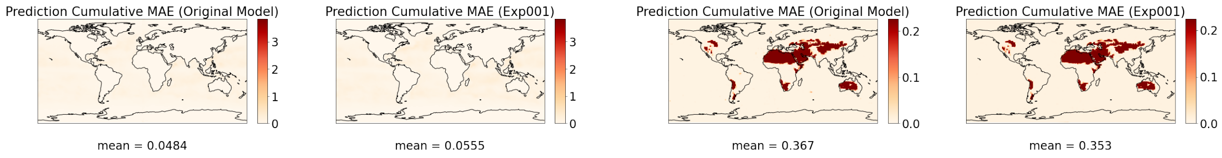
Figure 7.8: The difference maps of the yearly average for the prediction of the full model and for the prediction of the model trained on data with a different nature run.



(a) DMS (b) SO_2

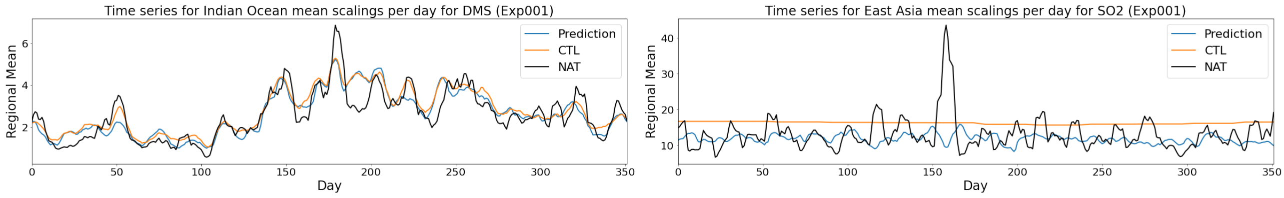


(c) BC (d) OC

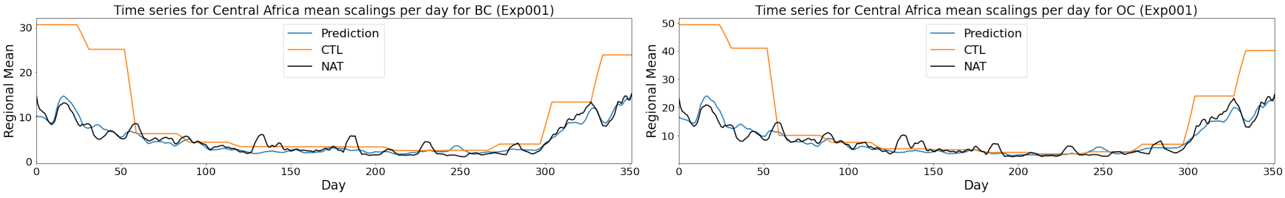


(e) SS (f) DU

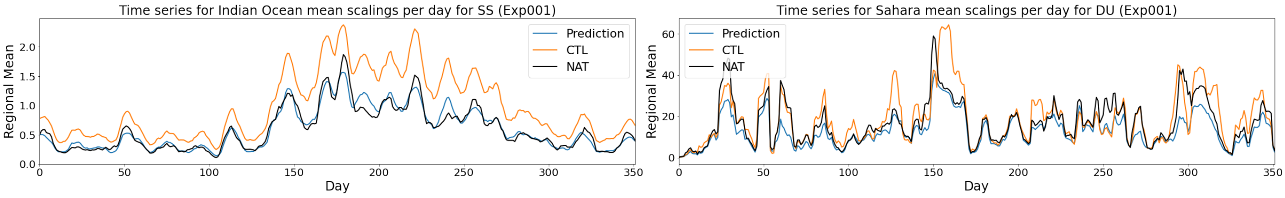
Figure 7.9: The cumulative MAE for the prediction of the full model and for the prediction of the model trained without SPEXone.



(a) DMS (b) SO_2



(c) BC (d) OC



(e) SS (f) DU

Figure 7.10: Time series of the regional mean for the region stated above each time series. The time series for NAT, CTL and the Exp001 model prediction are given for each species.

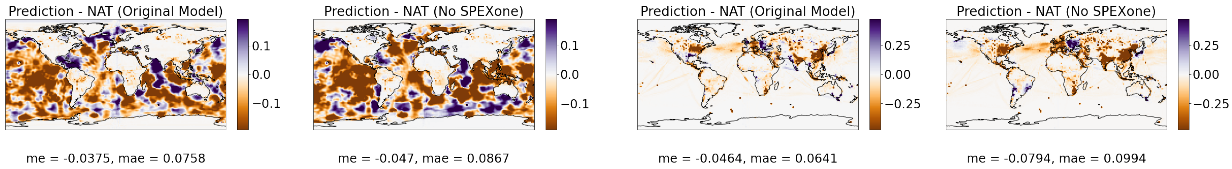
The difference maps in Figure 7.8 clearly show lower performance for Exp001 compared to the original model for all species. The regions where the model either underpredicts or overpredicts are larger for Exp001 for each species other than DU. This is also apparent in the MAE, where the original model has a lower MAE for all species with a decrease ranging between the 20% and 30%. Furthermore, the bias of Exp001 is higher, although it is still negative. Especially for OC, SS and DU we clearly see a higher negative bias. Also in Figure 7.9, we see a better performance of the original model for all species except DU. Exp001 has a slightly lower MAE_c mean, while it had a lower MAE over the yearly average. This is a result of the lower bias of the original model, resulting in more cancellations between under and overpredictions, which in turn cause a lower MAE of the yearly average. The other species have a lower MAE_c mean, with a decrease of approximately 10% compared to Exp001. The cumulative MAE of the original model for DMS shows clear differences to the cumulative MAE of Exp001. The area with a higher error over the Indian Ocean is smaller and also in the Pacific Ocean, the model has significantly lower errors.

The time series in Figure 7.10 look very similar to the time series from Figure 7.5, implying that both models perform about the same over regions with a lot of emissions. Nevertheless, the time series of the original model is slightly closer to the truth than the Exp001 time series. Especially for SO_2, we see a noticeable difference. This is possibly a result from the better performance of the model used for AOD, SSA an AE compared to CTL. In Section 6.2, we saw a big increase in performance of the model over East Asia, which is the region used for the SO_2 time series. Also BC, OC and DU show some small differences, although not as apparent as the differences for SO_2. The time series for both DMS and SS are show only very little differences. The reason for the similarity of DMS is possibly the low importance of DMS in aerosol measurements, as discussed previously. Therefore, a large increase in accuracy of AOD, SSA and AE for the input of the network does not translate to a large increase in accuracy for DMS. As for SS, the CTL fields for AOD, SSA and AE had a relatively lower error over the defined Indian Ocean, compared to some other defined regions. Consequently, the predicted AOD, SSA and AE fields for the input of the original model, do not offer as big of an increase over CTL for SS.

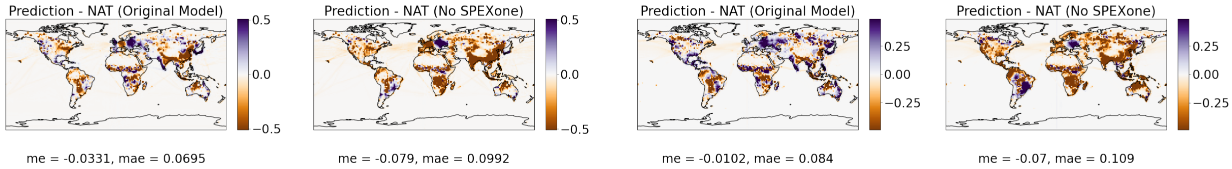
In Table 7.2, the average correlations of both the original model and the Exp001 model are given for each species in addition to the ME and MAE of the yearly average and the mean of the cumulative MAE. The table shows that the original model performs better than the Exp001 model for all metrics and for each species, with the exception of DU, where Exp001 has almost the same correlation and a lower MAE_c. It is also clear that Exp001 still provides a considerable improvement over CTL for all emission species other than DMS. The cause of the lack of performance increase for DU is likely due to the structure of the DU fields. The DU fields have a strong contrast between the areas where large emissions are present, like the Sahara, and areas without any emissions, which is the majority of the globe. This large contrast might cause problems for the neural network during training, especially since a large majority of the Earth does not have any emissions. Apart from DU, the original model has a clear improvement. Therefore, we can conclude that using the model for aerosol properties to obtain a global field of AOD, SSA and AE, which is closer to reality than the CTL fields, is beneficial for the performance.

7.3.2 Results from noSPEX

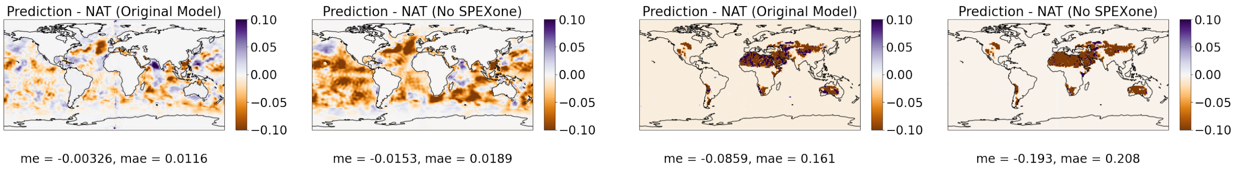
To test the importance of the inclusion of SPEXone as an input, we again perform another experiment where we train the same model, except we do not use SPEXone as an input and only use CTL fields. This means that the new input variables for AOD, SSA and AE are all $192 \times 96 \times 3$, however the rest of the network architecture stays the same. Since we only have one CTL field for the year, this also implies that CTL is the same for each ensemble member in the training data and for NAT in the testing data. As a result, one input corresponds to multiple different outputs, which is not ideal. The trained network without SPEXone, from now denoted as the noSPEX network, is first tested on the yearly average. The yearly average results on the validation set of the noSPEX model and the original full model are given in Figure 7.11. In Figure 7.12 the cumulative MAE for the original model and the noSPEX model are given and in Figure 7.13 the time series of the regional means are given. Lastly, the mean correlation, ME, MAE and MAE_c mean are given in Table 7.2 for CTL, the original model, Exp001 and noSPEX for all species.



(a) DMS (b) SO_2

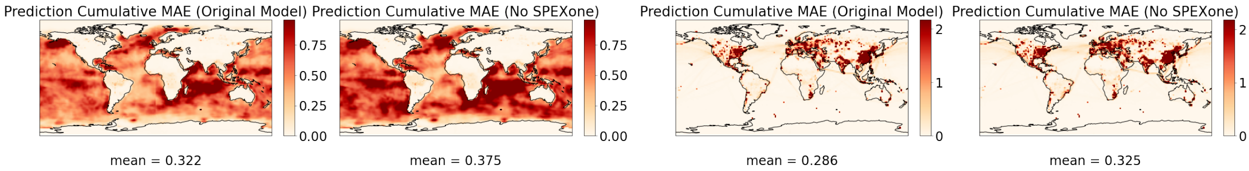


(c) BC (d) OC

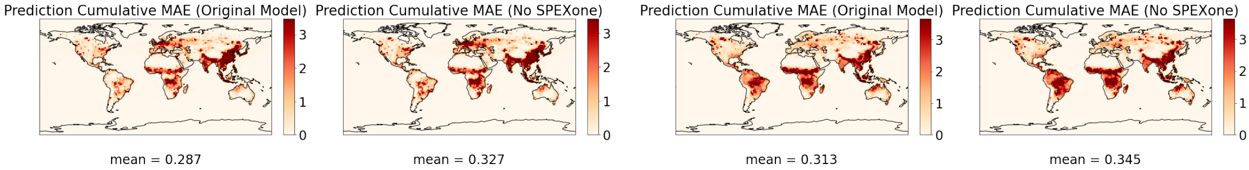


(e) SS (f) DU

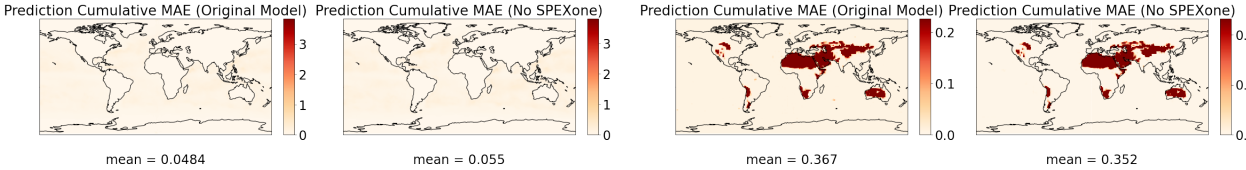
Figure 7.11: The difference maps of the yearly average for the prediction of the original model and for the prediction of the model trained without SPEXone.



(a) DMS (b) SO_2

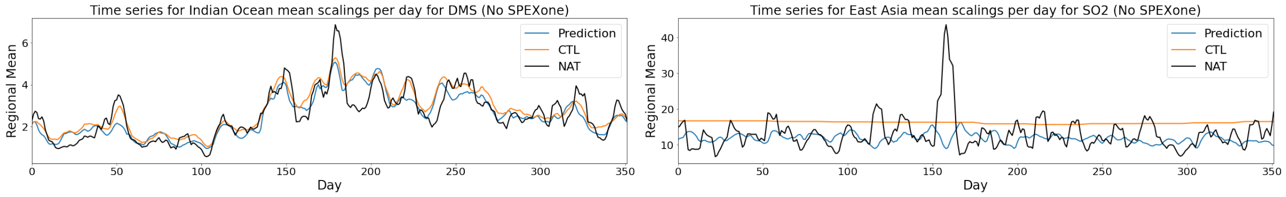


(c) BC (d) OC

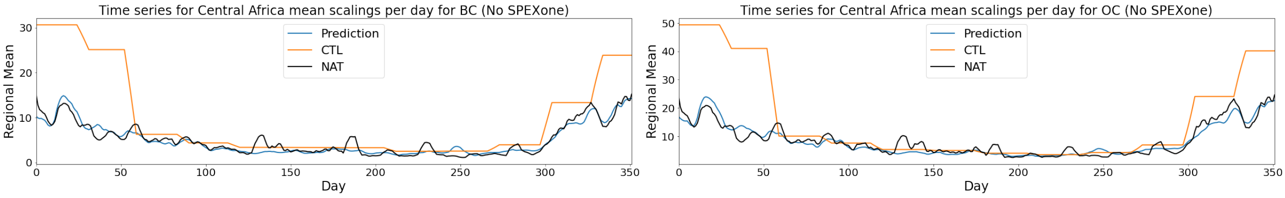


(e) SS (f) DU

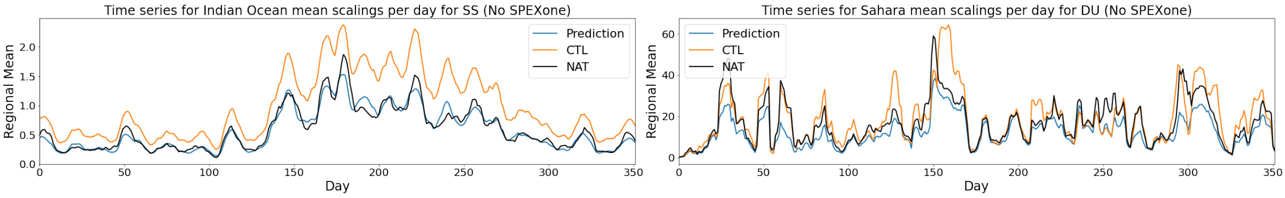
Figure 7.12: The cumulative MAE for the prediction of the original model and for the prediction of the model trained without SPEXone.



(a) DMS (b) SO_2



(c) BC (d) OC



(e) SS (f) DU

Figure 7.13: Time series of the regional mean for the region stated above each time series. The time series for NAT, CTL and the noSPEX model prediction are given for each species.

	Model	DMS	SO_2	BC	OC	SS	DU
Correlation	CTL	0.862	0.841	0.653	0.523	0.540	0.652
	Original Model	0.885	0.930	0.930	0.929	0.950	0.840
	Exp001	0.839	0.908	0.907	0.913	0.930	0.839
	noSPEX	0.840	0.908	0.908	0.915	0.929	0.839
ME	CTL	0.00698	0.106	0.123	0.0907	0.787	0.323
	Original Model	-0.0375	-0.0464	-0.0331	-0.0102	-0.00326	-0.0859
	Exp001	-0.0756	-0.0693	-0.0698	-0.0570	-0.0123	-0.171
	noSPEX	-0.0470	-0.0794	-0.0790	-0.0700	-0.0153	-0.193
MAE	CTL	0.0722	0.497	0.466	0.427	0.787	0.687
	Original Model	0.0758	0.0641	0.0695	0.0840	0.0116	0.161
	Exp001	0.101	0.0946	0.0935	0.107	0.0171	0.196
	noSPEX	0.0867	0.0994	0.0992	0.109	0.0189	0.208
MAE_c	CTL	0.356	0.620	0.637	0.668	0.791	0.811
	Original Model	0.322	0.286	0.287	0.313	0.0484	0.367
	Exp001	0.373	0.322	0.327	0.350	0.0555	0.353
	noSPEX	0.375	0.325	0.327	0.345	0.0550	0.352

Table 7.2: The average of the correlation time series, the ME and MAE of the yearly average and the mean of MAE_c for CTL, the prediction of the original model, Exp001 and noSPEX.

Comparing the figures and the results from the table, we notice that the original model again performs better, with the exception of DU. If we compare these results to the results from Exp001 in Section 7.3.1 or in Table 7.2, we observe that noSPEX performs similar to Exp001. There are very little differences in the global maps for both the yearly averages and the cumulative MAE maps. Also the time series show similar results, with a noticeably worse performance for SO_2 compared to the original model, some smaller differences for BC, OC and DU and almost no differences for DMS and SS. The metrics in Table 7.2 show no significant difference between Exp001 and noSPEX overall either, with the some exceptions like noSPEX having a lower MAE for DMS. The fact that noSPEX is still performing so well, practically the same as Exp001, means there are some problems in either the data or the network.

As explained previously, noSPEX was trained using only CTL fields, which means it had multiple outputs correspond to one input. Therefore, the model should not be able to reach such a high performance, especially considering it performs the same as Exp001, which had differing SPEXone inputs. The origin of this problem lies in the training and testing datasets. The training dataset was constructed from 8 different ensemble members, which therefore also results in little variance in training data as the ensemble members are still relatively similar to each other. Furthermore, the validation dataset was constructed from another ensemble member of the same run, which as a result was also close to the training data. Consequently, the model likely overfit, which also explains the same performance of Exp001 compared to noSPEX. Therefore, the difference between the inclusion of SPEXone and the absence of SPEXone might be due to the random weight initialisation of the network. For this setup, the sparse SPEXone data did not directly provide enough information to gain a better performing model. Luckily, SPEXone data does still provide extra information which can help obtaining better results, as we saw from the original model. If the SPEXone data is processed first, via the means of the neural network model for the aerosol properties problem, the network is able to extract enough information from it to get better predictions. Although the experiments performed on this data do not provide clear information on how well a network is able to learn the mapping from AOD, SSA and AE data to emissions, it does show it is able to predict emissions from accurate information on AOD, SSA and AE. This is seen in the better performance of the original model compared to Exp001 and noSPEX.

Chapter 8

Conclusions and Further Research

8.1 Conclusions and Discussion

In this report we proposed and constructed convolutional neural networks to solve two aerosol related problems. The first problem, the aerosol properties problem, was focused on obtaining a global field for AOD, SSA and AE from the limited SPEXone satellite observations and a control field with inaccurate values. The model network architecture proposed was a CNN like structure with two different encoders, one for SPEXone measurements and one for CTL, and with skip connections. For each AOD, SSA and AE a separate model was trained. All three models were trained for 200 epochs and converged. The trained networks were tested on the validation dataset against the results from NAT and CTL. The yearly average over the entire dataset was plotted and the ME and MAE over the yearly average were calculated. Both the ME and MAE were lower for the model compared to the ME and MAE of CTL for AOD, SSA and AE. In addition to the yearly average, we computed the cumulative MAE over the year. The models were again found to have lower errors than CTL, although the differences were not as big as previously. Lastly, the time series for the global means were compared against those of CTL and NAT and a time series for the correlation between the model and NAT was compared against a time series for the correlation between CTL and NAT. The model time series of the global mean followed the truth really well for AOD and AE, while it showed some clear deviations for SSA.

As for the correlations, the correlations of the three models were almost consistently higher than the correlations of CTL, which was also shown in a higher average correlation. Table **??** shows the average correlation, ME, MAE and MAE_c for AOD, SSA and AE. For all three cases we found that the model performed better than CTL for all four metrics. The model for AE performed the best with the highest average correlation and relatively the biggest decrease in MAE and MAE_c compared to CTL. SSA had the worst performance, with the lowest correlation and relatively the smallest decrease in MAE and MAE_c. This performance for each model is likely correlated to the performance of CTL itself, since CTL was used as an input. Since CTL has the highest correlation for AE and the lowest for SSA, this is a plausible explanation for the difference in performance. Another factor in performance is the random weight initialization of the network. If the weights are initialized in a favorable way such that the starting position of the loss is close to a global minimum, or low local minimum, the model can have better performance. Furthermore, the trained network for AE had a worse performance over the south pole region than CTL. The model had a clear overprediction, while the CTL AE was very close to the truth. This was likely due to a combination of high AE values around the south pole and a lack of SPEXone measurements in the region. Although the network was able to predict that there are high AE values near the south pole, it did not capture the exact values, since those were not available from SPEXone.

The second problem focused on training a neural network, with information for AOD, SSA and AE as input, to predict the emissions fluxes for the six emissions species DMS, $\mathrm{SO_2}$, BC, OC, SS and DU. The used AOD, SSA and AE fields were constructed using the trained network for the aerosol properties problem. These fields were averaged over 7 days, as well as the emission fields. The network was a CNN-like architecture with three encoders, for AOD, SSA and AE, six decoders, one for each species, and skip connections. The network was trained for 200 epochs and converged. The trained networks were tested on the validation dataset against the results from NAT and CTL. The yearly average over the entire dataset was plotted and the ME and MAE over the yearly average were calculated. We found that the model predictions had both lower ME's and MAE's than CTL for all species with the exception of DMS, where CTL had a significantly lower ME and a slightly lower MAE. Subsequently, the cumulative MAE over the year was computed, where we also found lower errors of the model than CTL, this time also for DMS. Time series of means over regions, with large emissions depending on the species, of NAT were compared against tge time series of the model and CTL. It was found that the model time series was closer to the NAT time series than the CTL time series was for all species. Especially the series for SO_2, BC and OC showed a big improvement over CTL, even though the prediction series of DMS, SS and DU were closer to the truth. Lastly, time series for the correlations between NAT and the model predictions and for the correlations between NAT and CTL were plotted and evaluated. The model had similar correlations as CTL for DMS, while it had consistently higher correlations for SO_2, BC, OC and SS. For DU, CTL had a higher correlation for some samples, however for most samples the model was higher correlated to NAT than CTL was. In Table 7.2 the average correlation, ME, MAE and MAE_c for all species are given. For all species, the model has a higher average correlation than CTL. The correlation is especially high for SO_2, BC, OC and SS. DU has the lowest correlation among the species, although it is still considerably higher than CTL. A possible explanation of the comparatively low correlation of DU is the structure of the DU emission fields. The vast majority of the globe does not have any DU emissions. The regions where DU emissions are present, the emission values are really high. This big contrast combined with the absence of emissions in most areas can cause trouble for the neural network during training. The average correlation of the model for DMS is higher than the average correlation of CTL, however the difference is not as big as for the other species. The other metrics also show little to no better results of the model than CTL. We do however note that CTL already has a high correlation and low errors for DMS, therefore it is more difficult for the network to provide an improvement. Furthermore, DMS is a very low contributor to the global AOD, which may also be a reason for the relatively small improvement, as it is harder to find a correlation between the input of the network and the DMS output.

8.1.1 Influence of SPEXone

For both the aerosol properties problem and the emissions problem, the network was retrained using data without SPEXone to test the importance of the measurements. The aerosol properties networks were simply retrained using the same CTL fields as input, however the second input of SPEXone was removed. The approach for the emission problem was slightly different, since the SPEXone information was already present in the AOD, SSA and AE fields. Therefore, for this setup, named noSPEX, we again resorted to using only CTL AOD, SSA and AE fields. For the emissions problem we then also trained a network, called Exp001, using these CTL fields in addition to unprocessed SPEXone measurements. The results of the models without SPEXone for the aerosol properties can be found in Table 6.3, Table 6.4 and Table 6.5, and the results of both alternate models for the emissions are shown in Table 7.2.

For the aerosol properties, all three models still performed considerably better than CTL if SPEXone was omitted and it was only trained on the CTL fields and their gradients. This could be due to some correlation between CTL and NAT. Alternatively, it could be a result from similarities between the training data and the testing data. However, the inclusion of SPEXone resulted in notably better results, with higher correlations and lower errors. This is expected as the SPEXone measurements contain direct information of the truth. Especially AOD and AE saw a big increase in performance by including SPEXone, with the results from AOD without SPEXone being almost the same as the CTL results. The increase in performance for SSA is the smallest. However, this is likely a result of the SSA values being very close to each other across the globe, therefore it is more likely for the network to make a small error which are considered bigger for SSA than for the network itself.

The results for the emissions showed that the model trained without SPEXone information performed worse than the original model for all species. Only DU had slightly better results when trained without SPEXone, possibly due to the structure of DU fields mentioned earlier. Again, noSPEX did improve over CTL like it did for the aerosol properties. When the results from noSPEX were compared with the results from Exp001 however, no significant difference was observed. Therefore, the model was able to learn the emission fluxes from just CTL, and SPEXone information did not provide extra information for training. This is a problem, since CTL does not necessarily contain any correct information, while SPEXone does. Furthermore, as the network was trained on 8 different ensemble members all with the same CTL fields, the network had multiple outputs correspond to one input. Since the models were tested on a different ensemble member not used for training, we expected that there was too little variation in the training data and the training data was too similar to the testing data. Consequently, the network likely overfit, which still provided reasonable results and no real difference between Exp001 and noSPEX. The better results from the original model compared to Exp001 and noSPEX still show that the network is able to extract information from SPEXone measurements, although they need to be processed first.

8.1.2 Final Conclusions

We conclude that it is possible to train a neural network to construct global AOD, SSA and AE fields from sparse SPEXone measurements. For all three properties the network was able to give accurate predictions, providing considerably better results than the CTL experiment. For emissions, a network is also able to provide improvements over CTL emissions. However, sparse SPEXone measurements do not directly provide enough information to the network and need to be processed first, for example by using a trained neural network to obtain globally covered fields. Furthermore, the network has trouble predicting emissions on an entirely different dataset from the training dataset. More testing needs to be done to give more sufficient conclusions on the exact capabilities of a neural network for emissions.

8.2 Further Research

To gain a better understanding of the capabilities of a neural network for emissions, more research needs to be done. Since the network in this report was trained on a dataset constructed from 8 different ensemble members, the training data had too little variation. We suggest to train the network on more and more varied data, for example by training it on 8 different years of data instead of 8 slight deviations of the same year. Furthermore, the model should be tested on another year of data as well, to get a more fair validation of the model. Since neural networks need a lot of different samples to learn the correct mapping from the input to the output, we expect this approach to give a better insight of the capabilities of a neural network.

Subsequently, the network can be trained for more emission species. This might however make the network too complex and too large, which can lead to computational issues during training. To circumvent this issue, multiple networks can be trained for different species.

Bibliography

[1] M. D. King, S. Platnick, W. P. Menzel, S. A. Ackerman, and P. A. Hubanks, “Spatial and temporal distribution of clouds observed by modis onboard the terra and aqua satellites,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 7, pp. 3826–3852, 2013.

[2] G. Myhre, C. Lund Myhre, B. H. Samset, and T. Storelvmo, “Aerosols and their relation to global climate and climate sensitivity,” *Nature Education Knowledge*, vol. 4, p. 7, 05 2013.

[3] O. P. Hasekamp, G. Fu, S. P. Rusli, L. Wu, A. Di Noia, J. aan de Brugh, J. Landgraf, J. Martijn Smit, J. Rietjens, and A. van Amerongen, “Aerosol measurements by spexone on the nasa pace mission: expected retrieval capabilities,” *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 227, pp. 170–184, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022407318308653

[4] A. Tsikerdekis, N. A. J. Schutgens, G. Fu, and O. P. Hasekamp, “Estimating aerosol emission from spexone on the nasa pace mission using an ensemble kalman smoother: observing system simulation experiments (osses),” *Geoscientific Model Development*, vol. 15, no. 8, pp. 3253–3279, 2022. [Online]. Available: https://gmd.copernicus.org/articles/15/3253/2022/

[5] N. Kriegeskorte and T. Golan, “Neural network models and deep learning,” *Current Biology*, vol. 29, no. 7, pp. R231–R236, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960982219302040

[6] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.

[7] D. Cornelisse, “An intuitive guide to convolutional neural networks,” https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/.

[8] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Prabhat, “Deep learning and process understanding for data-driven earth system science,” *Nature*, vol. 566, pp. 195–204, 2019.

[9] G. Camps-Valls, M. Reichstein, X. Zhu, and D. Tuia, “Advancing deep learning for earth sciences: From hybrid modeling to interpretability,” in *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, 2020, pp. 3979–3982.

[10] J. Y. Seo and S.-I. Lee, “Predicting changes in spatiotemporal groundwater storage through the integration of multi-satellite data and deep learning models,” *IEEE Access*, vol. 9, pp. 157 571–157 583, 2021.

[11] J. Zhu, S. Hu, R. Arcucci, C. Xu, J. Zhu, and Y.-k. Guo, “Model error correction in data assimilation by integrating neural networks,” *Big Data Mining and Analytics*, vol. 2, no. 2, pp. 83–91, 2019.

[12] J. Brajard, A. Carrassi, M. Bocquet, and L. Bertino, “Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the lorenz 96 model,” *Journal of Computational Science*, vol. 44, pp. 101–171, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877750320304725

[13] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, “Image inpainting for irregular holes using partial convolutions,” *ArXiv*, vol. abs/1804.07723, 2018.

[14] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[15] Q. Zhang, Q. Yuan, C. Zeng, X. Li, and Y. Wei, “Missing data reconstruction in remote sensing image with a unified spatial–temporal–spectral deep convolutional neural network,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 8, pp. 4274–4288, 2018.

[16] Y. Lops, A. Pouyaei, Y. Choi, J. Jung, A. K. Salman, and A. Sayeed, “Application of a partial convolutional neural network for estimating geostationary aerosol optical depth data,” *Geophysical Research Letters*, vol. 48, no. 15, 2021, e2021GL093096 2021GL093096. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2021GL093096

[17] A. Hernández-García and P. König, “Further advantages of data augmentation on convolutional neural networks,” in *Artificial Neural Networks and Machine Learning – ICANN 2018*, V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, Eds. Cham: Springer International Publishing, 2018, pp. 95–103.

[18] T. Haszpra, “Intricate features in the lifetime and deposition of atmospheric aerosol particles,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 7, p. 071103, 2019. [Online]. Available: https://doi.org/10.1063/1.5110385

[19] Delft High Performance Computing Centre (DHPC), “DelftBlue Supercomputer (Phase 1),” https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1, 2022.