Applying artificial intelligence to perform climate predictions

Data driven approach for parametrizing cloud cover

Hanna Svennevik

Thesis submitted for the degree of

Master of science in Computational Science

60 Credits

Department of Geoscience

Faculty of Mathematics and Natural Sciences

UNIVERSITY OF OSLO

August 3, 2020

Applying artificial intelligence to perform climate predictions

Data driven approach for parametrizing cloud cover

This work is published digitally through DUO Digitale Utgivelser ved UiO

http://www.duo.uio.no/Printed: Reprosentralen, University of Oslo

Abstract

There are large uncertainties in estimates of climate sensitivity for climate projections. These are related to cloud feedbacks and to how clouds are resolved in climate models. This inspire new parameterizations of cloud cover. It is unclear which level of sophistication is needed for these subgrid-scale parameterizations in order to model their effect on climate.

To investigate this problem from a new angle, namely the potential for data driven learning for parameterization of clouds, a new dataset, European Cloud Cover (ECC), was generated, using the self-implemented Area Weighting Regridding Scheme (AWRS). The dataset is a combination of reanalysis and satellite data, and it covers the period from 2004 to 2018.

Statistical methods, Autoregressive (AR)- and Convolutional Long Short-Term Memory (ConvLSTM)-models were trained and compared existing parameterizations in ERA5, on their ability to parameterize cloud cover.

Unfortunately, the $AR-B-L5$-model is unable to produce a realistic 24-hour forecast, as it suffers from "self-plagarims". Every prediction it generates is a small reduction of the cloud cover at the previous timestep.

$ConvLSTM-B_{10}-SL_{24}-32-3\times3-32-3\times3$, on the other hand, showed potential. A few hours into the forecast it is able to reproduce some of the features seen in ECC. Nevertheless, the model experienced some issues with blurred lines. $ConvLSTM-B_{10}-SL_{24}-32-1\times1-32-1\times1$ showed a lower skill on reproducing the test period from 2014 to 2018, but a superior ability to produce a 24-hour forecast.

None of the models in this study were able to compete with the state-of-the-art parameterization in ERA5. However, this study has successfully built a end-to-end trainable ConvLSTM-model on ECC and applied it to the cloud forecasting problem. Verifying the proof of concept that data driven learning on simple meteorological variables can be used to perform climate predictions.

Acknowledgement

This thesis is a joint work between University of Oslo (UiO) and Simula Metropolitan Center for Digital Engineering (SimulaMET). First and foremost I would like to thank my supervisors Hugo L. Hammer (SimulaMET) and Michael Riegler (SimulaMET) for suggesting this interesting topic of *Applying artificial intelligence to perform climate predictions*, and to Trude Storelvmo (UiO) for joining us, allowing me to shift the focus toward applications to cloud physics.

It is a poor workman who blames his tools, I admire how you tackled the corona pandemic and lead me safely to shore. Thanks for your patience, kindness and fruitful digital discussions, I've learned a lot from working with you!

I would also like to thank the scientist at EUMETSAT. Their expertise have been invaluable when generating the dataset.

A special thanks to Raymond Wiker, Jonah Shaw and Ina Storteig for their insightful comments. Your contribution have been valuable for my work.

A big thanks goes to my fellow master students, friends and family. Your support has been great. Finally, the constant encouragement of my sister, Sara, is sincerely appreciated.

Contents

List of Figures

List of Tables

Acronyms

AEROCOM Aerosols Comparisons between Observations and Models.

AI Artificial Intelligence.

ANN Artificial Neural Network.

AR Autoregressive.

AR5 Assessment Report 5.

AWRS Area Weighting Regridding Scheme.

BAMS Bulletin American Meteorological Society.

Batch Norm Batch Normalization.

Caltech California Institute of Technology.

CCN Cloud Condensation Nuclei.

CESM Community Earth System Model.

CFC Cloud Fractional Cover.

CMIP6 Coupled Model Intercomparison Project phase 6.

CNES National Center for Space Study.

CNN Convolutional Neural Network.

ConvLSTM Convolutional Long Short-Term Memory.

CRE Cloud Radiative Effect.

CRM Cloud Resolving Models.

DL Deep Learning.

ECC European Cloud Cover.

ECMWF European Centre for Medium-Range Weather Forecasts.

ECS Equilibrium Climate Sensitivity.

ERA5 ECMWF Reanalysis 5th Generation.

ESA European Space Agency.

ESM Earth System Model.

EUMETSAT European Organisation for the Exploitation of Meteorological Satellites.

eX^3 Experimental Infrastructure for Exploration of Exascale Computing.

GCM General Circulation Model.

GRIB GRIdded Binary or General Regularly-distributed Information in Binary form.

HRV High-Resolution Visible Channel.

IFS Integrated Forecasting System.

INP Ice Nuclei Particle.

IPCC Intergovernmental Panel of Climate Change.

IR Infrared.

JAXA Japan Aerospace Exploration Agency.

LSTM Long Short-Term Memory.

MAD Median Absolute Deviation.

MAE Mean Absolute Error.

MetNo Norwegian Meteorological Institute.

MFG METeosat First Generation.

ML Machine Learning.

MSE Mean Squared Error.

MSG METeosat Second Generation.

MWIR Mid-Wavelength Infrared.

NASA National Aeronautics and Space Administration.

netCDF network Common Data Form.

NIR Near-Infrared.

NorESM Norwegian Earth System Model.

NWP Numerical Weather Prediction.

PDF Probability Density Function.

RDMA Remote Direct Memory Access.

RGB Red-Green-Blue.

RNN Recurrent Neural Network.

SEVIRI Spinning Enhanced Visible and Infrared Imaging.

SimulaMET Simula Metropolitan Center for Digital Engineering.

STD Standard Deviation.

SWIR Short-Wavelength infrared.

TOA top-of-the-atmosphere.

UiO University of Oslo.

VIS VISible spectrum.

WV Water Vapor.

1 Introduction

Weather and climate have a major influence of life on Earth. While the globe struggles to support the needs of an increasing population, human activity continues to change the environment.

Technological advances starting with the industrial revolution have caused a steady increase in anthropogenic emissions since the 18^{th} century. This has lead to a increase in the atmosphere CO_2-concentration. CO_2 is a greenhouse gas and the emissions cause a imbalance in the energy budget. Greenhouse gases cause a decrease in longwave radiation emitted to space, driving global warming. Greenhouse gas emissions have a delayed effect on temperature and it will keep rising until it has reached the equilibrium temperature, closing the energy budget.

Providing a realistic estimate of future climates is important for motivating mitigation and limiting the global temperature increase. This introduces the need for studying potential futures using a combination of models and idealised experiments. A common, yet not very intuitive quantity used to describe this temperature increase is the Equilibrium Climate Sensitivity (ECS). The ECS is computed as the linear fit between perturbed global mean surface temperature and the radiative imbalance at top-of-the-atmosphere (TOA).

Estimates based on simulations prepared for the latest assessment report (AR5) produced by the Intergovernmental Panel of Climate Change (IPCC) puts ECSs in the range from 2.1^oC to 4.7^oC (Flato et al., 2013, p.817). These runs are forced with instantaneous quadrupling of the CO_2-concentration and keeping it constant for 150 years. AR5 was published in 2013, since then a lot of research efforts have been invested in reducing the spread (Cox, Huntingford, and Williamson, 2018). The socio-economic consequences related to the uncertainty is enormous (Bony et al., 2015).

Including the effects of clouds on large scale processes in climate models, has proven difficult. The Flato et al. (2013) associate a high confidence to the claim stating that uncertainties attributed cloud processes explain much of the spread in modelled climate sensitivity.

After decades of being mainly a research area of limited use, today Artificial Intelligence

(AI) in the form of Deep Learning (DL) has become a part of our daily life. Face recognition technology has the ability to unlock your phone. Speech recognition allows you to dictate text messages. Self-driving buses are used in metropolitan areas around the world, including Oslo city centre. Image manipulation allows you to transform Monet paintings to pictures and back (Zhu et al., 2017). Video manipulation can make people appear to say and do things they never did.

DL has already proven its value in many fields. The scope of this study is to implement and compare different methods for data driven learning to find the most suitable method for cloud fractional cover predictions. For this task Autoregressive (AR) models and Convolutional Long Short-Term Memory (ConvLSTM) are considered.

Compared to existing cloud parameterizations implemented in General Circulation Model's these models provide a substantial simplification. However, this simplification is requirement to enable the use of DL. This study attempts to answer if it is an appropriate method for solving cloud cover predictions. Occam's razor is an old philosophical concept. It states that if two hypotheses are equally likely, the simplest one should be preferred (Duignan, 2020).

For this application, the compilation of a new dataset was deemed necessary. The dataset is named European Cloud Cover (ECC), and is composed of reanalysis data from ERA5 (Hans Hersbach et al., 2020) and satellite retrievals from METeosat Second Generation (MSG) (Schmetz et al., 2002). The dataset has a temporal resolution of one hour and spatial resolution of 0.25^o. Clouds have an average lifetime of less than one hour (Lohmann, Lüönd, and Mahrt, 2016, p. 19), thus it should be achievable to extract suitable features to make reasonable predictions.

There is currently considerable interest in and activities around Machine Learning (ML) and AI. The practical applications to climate research are under investigation. The last few years researchers have attempted to incorporate DL into Geosciences, with the goal of solving a wide range of problems, from rainfall runoff modeling (Kratzert et al., 2019) to high-resolution weather forecasting (Rodrigues et al., 2018). Another more comprehensive ML project is led by Tapio Schneider at California Institute of Technology (Caltech). Along with his team he has ambitions to create a next-generation Earth System Model (ESM) using ML, by creating a platform that can resolve clouds and hopefully reduce the uncertainty in climate

sensitivity (Voosen, 2018).

There is high confidence that a lot of the spread in ECS can be attributed to cloud feedbacks (Flato et al., 2013,p.817). Developing new methods for cloud cover predictions is believed to produce more reliable estimates of future climates. Focusing on a simplified representation that still captures the most relevant processes stands the best chance of reducing the spread. This thesis is a "proof of concept" study investigating the potential for using a data driven approach for parameterizing Cloud Fractional Cover (CFC) based on standard environmental variables.

2 Climate and Cloud Physics

This chapter describes the necessary theoretical background on clouds for this study. It is organised as follows. The three first sections describe the cloud's role in both present and future climate systems. Beginning with cloud formation and dissipation, including the cloud effects on Earth's radiative budget in the current climate and the suggestion of future cloud climatologies as proposed by IPCC in AR5. This is followed by a brief introduction to parameterization of clouds, i.e. the method used to incorporate clouds into climate models. Finally, the data used in the compilation of the ECC dataset is introduced.

2.1 Clouds role in the climate system

Clouds play an important role in the climate system both affecting the radiative budget and the hydrological cycle. Understanding how clouds form in the complex system of the atmosphere involves both knowledge about the large scale influence by the circulation and the small scale influence by aerosols. Clouds exist in countless number of shapes and sizes, and have fascinated mankind since the beginning of time. Figure 2.1 shows the stunning view from Store Smørstabbtinden in Jotunheimen. The sky is covered by cumulus type clouds, a common sight in summer.

Figure 2.1: *Cumulus deck at Store Smørstabbtinden in Jotunheimen, photo by Ina Storteig.*

2.1.1 Evolution of clouds

Clouds are composed of liquid droplets, ice crystal or both. To this day the microphysics of all phases are not fully understood. Here mixed phase clouds, consisting of both liquid and ice, have proven to be the most difficult to fully understand (Boucher et al., 2013).

Aerosols include both liquid and solid particles suspended in the air. They interact with the clouds by serving as particles which vapor and ice can condensate or deposit upon. The different phases require different properties and the nuclei are called Cloud Condensation Nuclei (CCN) for liquid droplets and Ice Nuclei Particle (INP) for ice crystals. In the following discussion, *saturation* describes the equilibrium state between to phases. For phases such as liquid water and vapor, saturation implies equal rates of condensation and evaporation. Phase changes occur when the system deviates from the equilibrium state. Under supersaturated conditions, the rate of condensation exceeds the rate of evaporation, facilitating vapor to condense onto suitable aerosols and initiating the formation of clouds.

Saturation is usually achieved by a temperature decrease in rising air masses. The saturation vapor pressure, e_s, is the quantity describing the maximum amount of vapor air can retain at a certain temperature. The way in which e_s depends on the temperature, T, is described by the Clausius-Clapeyron equation for water, see Equation (2.1). The entalphy of vaporization, l_v, is the amount of energy needed to evaporate one unit of molecules (e.g one mole) from the liquid. This is also known as the latent heat of vaporization.

$$\frac{de_s}{dT} = \frac{l_v e_s}{RT^2} \tag{2.1}$$

Here $l_v = 40.8 \cdot 10^3 Jmol^{-1}$ and the universal gas constant $R = 8.314 Jmol^{-1}K^{-1}$ (Lamb and Verlinde, 2011, p. 42).

A solution to Equation (2.1) is given in Equation (2.2). It is derived by integrating from $T_0 = 273.15K\,(0°C)$ to an arbitrary temperature, T. The integral is intractable for varying l_v. However a constant l_v is a reasonable assumption for the ranges of temperatures of atmospheric interest. The lower boundary, T_0, is chosen based on convenience, motivated by the fact that the constant of integration, e_0, needs to originate from measurements. At T_0, the equilibrium of a mixture of water and ice at a total pressure of 1 atm is $e_0 = 611Pa$.

$$e_s(T) = e_0 e^{\frac{l_v}{R}\left(\frac{1}{T_0} - \frac{1}{T}\right)} \tag{2.2}$$

From Equation (2.2) it is clear that e_s increases with rising temperature, resulting in the phenomena that warmer air can retain more vapor. The same principles apply for the phase change sublimation, but its entalphy, l_s, has a distinct value (Lamb and Verlinde, 2011, p.135). The saturation vapor pressure with respect to ice, e_i, can be derived by replacing l_v by l_s. Subsaturated conditions cause the cloud liquid water to evaporate, and ultimately the cloud disappears. Often caused by a mixing with dry ambient air or a temperature increase.

Growth processes are phase dependent. Liquid droplets grow by diffusion and later by collision and coalescence. At temperatures around -38 ^oC (Lohmann, Lüönd, and Mahrt, 2016, p.222) droplets spontaneously freeze, while at warmer temperatures freezing can only occur with the aid of an INP. Clouds consisting purely of ice crystals first grow by deposition of vapor and then by aggregation (Fowler and Randall, 1996). In the presence of both phases, the Wegeron-Bergeron-Findeisen process describes the mechanism where droplets evaporate and the vapor deposits on to the ice crystals. This mechanism exists because the saturation vapor pressure is lower with respect to ice than water, $e_i < e_s$. The process is most efficient at -12$^\circ C$ when the difference is largest. Macrophysical properties describe the clouds as units, using properties like base height, top height, thickness, fractional cover and regime (also known as type). Microphysical processes are all mechanisms involving the particles forming a cloud. Examples of properties used to quantify the microphysical state are CCN and droplet number concentrations (Grabowski et al., 2019).

2.1.2 Clouds role in the radiative budget

The characteristic white colour of the clouds has it nature in its ability to effectively scatter solar radiation. The Earth bathes in radiation from the Sun. Passing through the atmosphere, a small portion of the radiation gets absorbed while another portion gets scattered by clouds and aerosols. The majority of the radiation reaches the Earth and transforms into heat, warming the surface. The Earth emits thermal radiation, a minor portion of which escapes directly back to space, while most of it gets absorbed by the atmosphere and is re-emitted. This phenomena is known as *the greenhouse effect*.

The amount of heat trapped in the Earth system depends fundamentally on the spectral properties of its components (i.e. clouds, greenhouse gases, aerosols), and determines the magnitude of the enhanced warming (Zhong and Haigh, 2013).

Albedo is the ratio of reflected to incoming radiation. Dense low level clouds have high number concentrations of droplet, which corresponds to a large surface area. This results in enhanced scattering of radiation and thus a higher albedo. The greenhouse effect of clouds follows the principals of the greenhouse effect described above. It arises from their ability to absorb thermal radiation and re-emit it. The absorbed radiation originates from the surface or the atmosphere below. A widely used assumption is that the Earth (and most clouds) radiate like a black body, thus its radiant flux is given by Stefan-Boltzmann fourth-power law,

$$F = \sigma T^4 \tag{2.3}$$

here F denotes flux in units of Wm^{-2}, T denotes temperature in units of K and $\sigma = 5.670^{-8} Wm^{-2}K^{-4}$ is the Stefan-Boltzmann constant.

Mediums like water, snow and ice are not necessarily perfect emitters, this requires the need for modifying Equation (2.3) with a scaling factor, called emissivity, $\varepsilon \in [0,1]$, this depends on the composition and density of the medium. The emitted flux is given by $F = \sigma\varepsilon T^4$. This provides an additional source of uncertainty to the computations of the greenhouse effect of clouds and therefore also the ECS.

Researchers are still struggling with determining the exact spectral emissivity of different mediums. This is of interest for both implications to the radiative transfer calculation, but it is also of utmost importance in the field of remote sensing, where distinguishing the signal from the reference signal continues to pose as a problem. The greenhouse effect increases with the cloud altitude, enhanced by the increased temperature difference between the surface and cloud. High clouds with low temperatures re-emit radiation at a lower intensity than they absorbed. Energy thereby gets trapped in the Earth system, which has a warming effect.

2.2 Clouds in the current climate

On the basis of simulations and available observational data, both remote sensing and in-situ measurements, Wild et al. (2019) have quantified the contribution of elements in the Earth's annual global mean energy budget. The Cloud Radiative Effect (CRE) is computed by subtracting the components of a cloudy atmosphere from a cloud-free atmosphere (Ramanathan et al., 1989), usually at the top-of-the-atmosphere (TOA). The altitude along with the composition of clouds determine their optical properties and in turn their interactions with radiation.

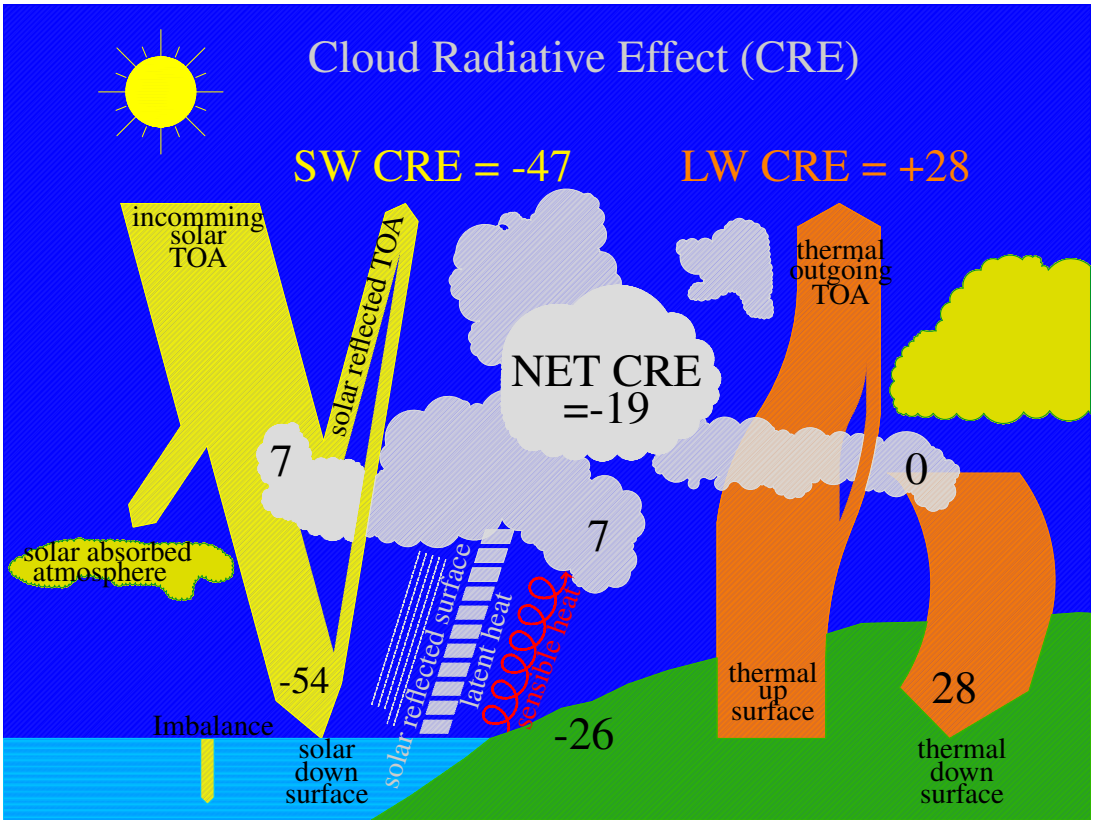


Figure 2.2: *The global mean annual Cloud Radiative Effect (CRE) is the difference between the radiative components of the clear-sky (cloud-free) and all-sky (cloudy) radiative components. A positive sign can be describes a warming effect and negative a cooling, units are in* Wm^{-2}*. Inspired by Figure 15 in Wild et al. (2019) .*

Figure 2.2 shows a schematic illustration of the CRE in the Earth's TOA annual mean energy budget, a negative sign denotes a cooling effect and a positive sign can be associated with a warming effect. Wild et al. (2019) find a reduction in incoming solar radiation of $-47Wm^{-2}$ caused by clouds, showing that clouds reflect approximately 14% of the incoming solar radiation.

The thermal CRE amounts to $28Wm^{-2}$, resulting in a net CRE of $-19Wm^{-2}$. This proves that the net effect of clouds on the TOA radiative budget is negative, and that clouds currently have a cooling effect on the climate. For the details on the all-sky (cloudy) and clear-sky (cloud-free) energy budgets, used in the computations of the CRE, please see the paper Wild et al. (2019) .

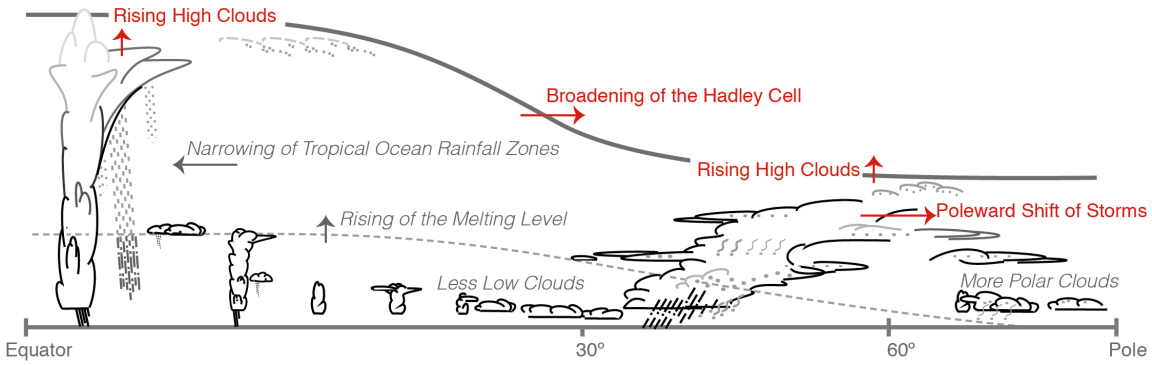


Figure 2.3: *Expected cloud changes in future climate. This figure was developed by IPCC based on feedbacks in climate models, and the different adjustments are associated with different levels of confidence. (Boucher et al., 2013).*

2.3 Clouds in future climates

As concluded in the previous section, an excess of radiation currently gets trapped in the Earth system, forcing the atmospheric temperature to increase in order to ultimately close the radiative budget. The temperature increase induces climate change and recent estimates find the current imbalance at TOA to be $0.6 W m^{-2}$ (Wild et al., 2019). *Global radiative equilibrium* is reached when the temperature of the atmosphere is adjusted such that the radiation emitted to space is equal to the portion absorbed by the surface.

This heat gets trapped in the Earth system, forcing the surface temperature to increase in order to close the radiative budget. The imbalance in the radiative budget at top-of-the-atmosphere (TOA) is the radiative forcing. Climate drivers include both natural and anthropogenic forcings. A *forcing* can be everything from natural variability in the solar energy output, volcanic eruptions or greenhouse gas emissions. The climate science community works toward a common goal to determine the magnitude of the forcings responsible for the observed climate change since pre-industrial times, and the associated climate response as quantified by the ECS. The latter is controlled by climate feedback processes, of which those associated with clouds are the most uncertain.

Figure 2.3 shows a summary of the most likely cloud feedbacks and the shift in cloud regimes suggested by the IPCC (Boucher et al., 2013, pp.591-592).

First, a broadening of the Hadley cell causes a poleward shift of storms. This dries the subtropics and moistens the higher latitudes. Northward propagating clouds cause a reduction

in the albedo effect. The radiation available for reflection decreases poleward, disappearing into the polar night, as a direct consequence of the Earth's spherical geometry. The greenhouse effect of clouds still persist without sunlight leading to a heating due to clouds in the Arctic, in contrast to their global effect (Wang and Key, 2005). Second, rising high clouds motivate a stronger greenhouse effect. Third, a reduction in the presence of low level clouds reduces the amount of reflected solar radiation. The reduced reflection of solar radiaton is assumed to be partly offset by an lifting of the melting layer. Consequently, ice crystals are replaced by liquid droplets and the phase transition results in more opaque clouds (Boucher et al., 2013, p.592)

2.4 Parametrization of Clouds

All global climate simulations are limited by computational power and the typical model grids are much too coarse to resolve all relevant processes governing clouds. Parameterization allows us to nevertheless simulate the effects of clouds on the climate, through simplified representations of cloud processes that are a function of resolved model variables. The development of both observational and modeling systems requires an understanding of the physical and biogeochemical processes that take place in the Earth system (Simmons et al., 2016). New descriptions of mechanisms are implemented into models and tested against observations. The simulations should be able to recreate previous climate.

The complex nature of clouds originates from lots of different processes occurring simultaneously on different scales. Incorporating all these interactions into a model framework has proven to be difficult (Boucher et al., 2013, pp. 584)

2.4.1 Relative Humidity and Statistical Schemes

The simplest form of cloud scheme is a binary. A model grid box is either cloudy or clear. Equation (2.4) describes a diagnostic relationship between cloud cover and relative humidity. Binary saturation threshold can be implemented as follows,

$$CFC(RH) = \begin{cases} 0, & \text{if RH} \leq 100 \\ 1, & \text{else} \end{cases} \tag{2.4}$$

Many high-resolution model apply this approach, but it is not suitable for ESM having spatial resolution in the scale of 100km (Tomkins, Adrian M, 2005).

Most climate models have a cloud fractional cover, which is driven by a saturation thresh-

old. All the vapor in excess of this threshold, often $RH = 100\%$, gets transformed into cloud liquid water. An assumption of sub-grid scale variability is necessary to achieve fractional cloud cover. The most common variables either alone or in combination are relative humidity, temperature and vertical velocity (Golaz, Larson, and Cotton, 2002a). Using a fixed threshold for the critical relative humidity is a necessary, but rough simplification.

In statistical schemes relative humidity and other dependent variables are simulated using Probability Density Function (PDF). The distribution are difficult to obtain theoretically. A common approach is to draw these distributions empirically, based on observations from weather balloons or airplane campaigns. In this way the functional form of the underlying distributions have a physical basis. Observations have been made during varying cloud conditions and almost all existing PDFs have been used in statistical cloud schemes. The parameterization is then very sensitive to the choice of moments, i.e. mean, variance, skewness and kurtosis. Employing fixed moments in a statistical scheme simplifies to relative humidity schemes (Tomkins, Adrian M, 2005). Researchers have not been successful in finding an adequate representation of cloud cover using these approaches (Adrian M Tompkins, 2009).

Golaz, Larson, and Cotton (2002) derived a joint PDF of the subgrid-scale variability, serving as the base for parameterizing boundary layer clouds. This scheme is implemented in Norwegian Earth System Model (NorESM) (Seland et al., 2020) and Community Earth System Model (CESM) (Danabasoglu et al., 2020), two recognized ESM. The parameterization can be considered a higher-order turbulent closure problem. The first (mean), second (variance) and third (skewness) order statistical moments, of the vertical velocity (w), the liquid water potential temperature (θ_l), and the total water specific humidity (q_t) determines the family of PDFs. It is designed to be flexible enough to circumvent the case specific adjustment (Golaz, Larson, and Cotton, 2002a, Golaz, Larson, and Cotton, 2002b).

2.4.2 Cloud resolving models

Another method of cloud parameterization is deriving relationships from Cloud Resolving Models (CRM)-models. These relationships again be implemented into global climate models. Contrary to what the name implies, this type of model still has problems with resolving the very smallest cloud processes, occurring on micrometer-scales. CRM's are computationally expensive and can only be run for a short amount of time. One weakness of

this approach is the dependancy of the CRM-models own parametrizations of microphysics (Tomkins, Adrian M, 2005).

2.4.3 ECMWF IFS-Model

European Centre for Medium-Range Weather Forecasts (ECMWF)s numerical weather prediction model is named, Integrated Forecasting System (IFS). Like all operational systems, the model is continuously improved. The complexity of the cloud cover parameterization is described below in non-mathematical terms.

Today's implementation is heavily inspired by the prognostic scheme for stratiform and convective cloud for large-scale models, published in 1993 and used operationally since 1995 (Tiedtke, 1993, Tomkins, Adrian M, 2005). Forbes and A. M. Tompkins (2011) upgraded the representation of clouds and precipitation in the IFS, cycle (36r4). They extended the number of prognostic variables cloud fraction, cloud liquid water, cloud ice, rain and snow. This improves the physical basis of the moist microphysical schemes and also makes the model more suitable for varying resolutions. Its now more like a high-resolution limited-area NWP models and CRM (Forbes and A. M. Tompkins, 2011). The scheme also considers cloud formation by large-scale ascent, diabatic cooling, boundary-layer turbulence, advection of cloud water from convective updrafts. Cloud dissipation caused by adiabatic and diabatic heating, mixing with environmental air and depletion of cloud water. It can also generate anvils, allow cirrus clouds to form from convective updrafts and boundary layer clouds (Tiedtke, 1993).

2.5 Data

This section presents the data used in the compilation of the dataset, including background information about remote sensing of cloud properties and the method used to retrieve the data.

2.5.1 ERA5

ERA5 is the latest in the series of reanalyses produced by European Centre for Medium-Range Weather Forecasts (ECMWF). Reanalysis is as close to observations as one can get while still obtaining data that is complete and coherent in both space and time. It is produced using a forecast model to assimilate observations. Data assimilation take observations as input and tries to make an accurate estimate of the state of the system that is as consistent as possible with the available observations at all times. This includes observations retrieved

from satellites, ships, buoys, airplanes and ground-based stations. The analysis is produced in the operational forecast system, making it available within five days of real time. ERA5 is based on the IFS, cycle 4lr2. The data is available at a horizontal resolution of 0.25^o degree and hourly temporal resolution. It is an important product for the continuous monitoring of the Earth system (Hersbach et al., 2018). Some of the variables in ERA5 is assimilated, others are parameterized based on the assimilated variables, e.g. cloud fractional cover.

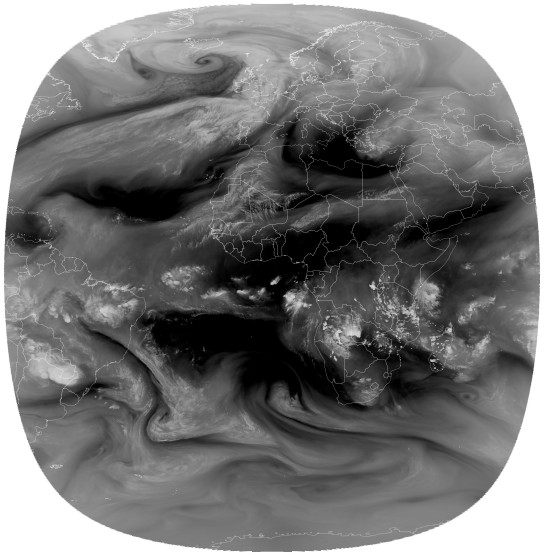
Reanalyses data is often mistakenly referred to as observations. Parker published an essay in Bulletin American Meteorological Society (BAMS) on this topic in 2016. Based on the following three points they conclude that observations and reanalyses are not too different. First, both involve inference, in other words, theory-based calculations. Second, reanalysis relies on forecast and observations do not. This is not a significant difference as long as the forecast is sufficiently accurate. Third, it is important to be aware that the uncertainty of the reanalyses is less well known than for observations. This makes it harder to judge the appropriate use of the reanalyses (Parker, 2016).

2.5.2 Remote Sensing of Cloud Properties

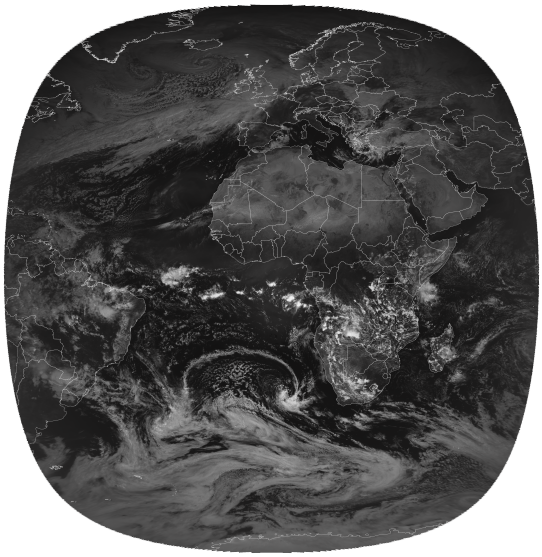
Satellites are the only instruments capable of providing continuous global measurements. Measurements are collected by sensors, of which two types exist; passive and active imagers. The passive imaging sensors detect natural occurring levels of radiation, e.g. thermal radiation emitted by mediums. In contrast, active sensors detect radiation returned from an emitted artificially fixed pulse of radiation (Stephens et al., 2018).

Karlsson, Johansson, and Devasthale (2015) list the five key properties exploited in remote sensing of clouds using passive imagery, these will be explained drawing examples from Figure 2.4. High radiances are displayed in white and lower radiances in darker colors. In general anything that appears bright has a higher reflection at the TOA than the surface.

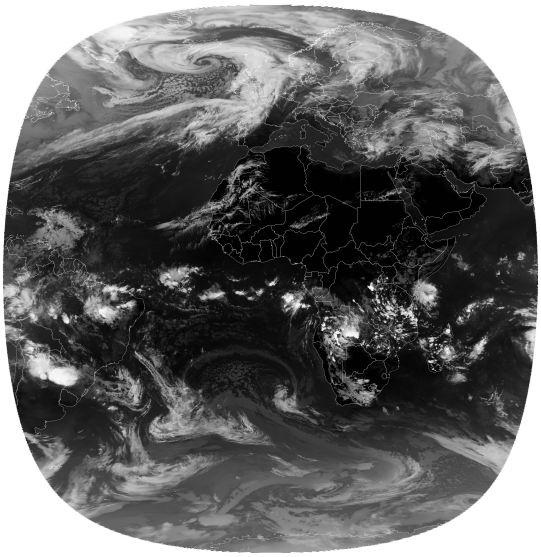
Firstly, clouds appear bright as opposed to ice free water surfaces and vegetated Earth's surface in VISible spectrum (VIS) and Near-Infrared (NIR). The detection from VIS 0.6 is shown in Figure 2.4b. Secondly, clouds consisting of liquid droplets reflect strongly in Short-Wavelength infrared (SWIR) and Mid-Wavelength Infrared (MWIR). The signal from MIR 3.9 is displayed in Figure 2.4d. Here the Earth's surface, including snow and ice, appear dark. Clouds are not perfectly emmiting black bodies and this allows for detection of low level clouds at night.



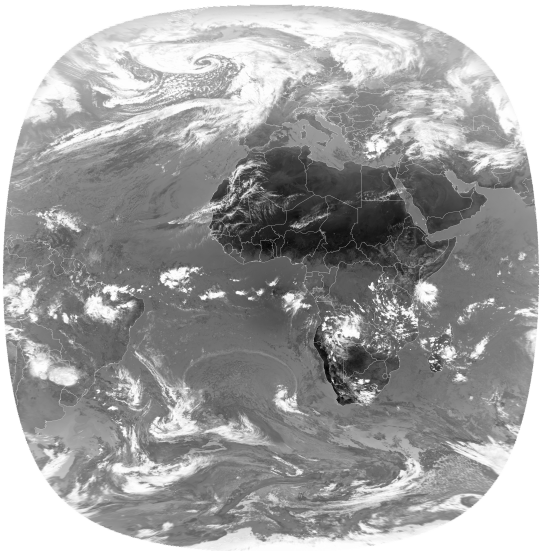
(a) Channel WV 6.2



(b) VIS 0.6



(c) IR 10.8



(d) MIR 3.9

Figure 2.4: *Spectral bands from SEVIRI. February 15, 2020 at noon. It shows the low-pressure system Elsa positioned of the west coast of Iceland. Having a record breaking low of 915hPa (Forland, 2020). The images are provided by the EUMETSATs image archive (EUMETSAT, 2020).*

The third property exploited is that clouds are typically colder than the Earth's surface. Therefore they appear bright in IR channels, when displayed in reverse i.e. with low radiences shown as bright.

The fourth property is that cirrus cloud are optically thin, but can be detected using split

window channels (IR10.8 and IR12.0) differences (see Figure 2.4c). The last property is not shown in this figure, but it is that in general broken clouds typically give rise to scattered patterns or texture in images with otherwise homogeneous, ice-free ocean for instance.

Figure 2.4a also show the signal detected from WV 6.2, water vapor have a absorption band at these wavelength useful for detected water vapor.

To summarize, the success of a screening is dependent on the illumination, the state of the surface and atmosphere. And it has proven most difficult over bright surfaces like snow and dessert (Karlsson, Johansson, and Devasthale, 2015).

Improved technologies allow for measuring new variables, one example of such advances is the use of active sensors. Active sensors provide a more refined image of vertical profiles and allows for the detection of cloud phase. Despite the limitations of passive sensors they provide useful historical information and in some cases higher spatial and temporal resolutions. One attempt to relate passive measurements with active measurement under the same meteorological conditions is the *Afternoon constellation, A-Train* launched as a cooperation between several space agencies; National Aeronautics and Space Administration (NASA), Japan Aerospace Exploration Agency and the french National Center for Space Study (CNES). This has provided a useful link between the different types of measurements (Stephens et al., 2018).

The number of spectral bands and footprint size (pixel resolution) determine the practical application of a particular satellite retrieval. Differences in swath width determine the frequency at a given position, determining the temporal resolution. The viewing angle affects the optical properties of the medium, and also its apparent position. *Parallax* describes the apparent shift in position of a object, caused by moving the observer along an axis. This is a known issue in remote sensing. Positions of satellites are changed. Moving the observer (satellite) changes the apparent position of the measurement. High viewing angles may cause similar issues, in detection of high clouds. This factor is negligible when detecting low clouds (Joro, Hyvärinen, and Kotro, 2010).

Differences in sensitivity and retrieval algorithms contribute to a large spread in global mean cloud amount among different cloud products. This also explains the difficulties involved in using one cloud dataset to evaluate another one. In their assessment of global cloud datasets Stubenrauch et al. (2013) compared the global mean cloud cover of six datasets (ISCCP,

PATMOS-x, MODIS-ST, MODIS-CE, AIR-LMD and TOVS Path-B). In this process they eliminating MISR and ATSR-GRAPE because of different observation times (see Table 3 in Stubenrauch et al. (2013)) and two outlier datasets, HIRS-NOAA and POLDER. Their results show that the difference among the six datasets is of order 0.08. In contrast, local differences could be up to 0.4 (Stubenrauch et al., 2013).

2.5.3 METeosat Second Generation, MSG

The METeosat Second Generation (MSG) was established as a corporation between European Space Agency (ESA) and European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT). ESA was in charge of developing the prototype of MSG-1. EUMETSAT is responsible for maintaining the user requirements, launch procedures, developing ground segments, ensuring overall system consistency and day to day operations.

The primary function of MSG is to provide continuous observations of the Earth's full disk. Near-constant sampling frequency and a geostationary orbit allows for observing weather phenomena occurring on short timescales. Located at 0^o MSG provides a varying spatial resolution, unlike the polar orbiting satellites. The resolution gets coarser with increasing off-nadir viewing angle (Stubenrauch et al., 2013).

There are efforts invested in extending the MSG dataset with the METeosat First Generation (MFG), in order to make use of the time series all the way back to 1980 (Bojanowski et al., 2018). This requires new cloud detection algorithms since they only have three common channels and only two of them are useful for detecting clouds (Stöckli et al., 2019).

On board the MSG is the Spinning Enhanced Visible and Infrared Imaging (SEVIRI) imaging radiometer. It has 12 spectral channels. The scan is done south to north, east to west. The wavelengths of the discrete channels are chosen based on heritage from other sensors. seviri has one broadband visible channel, three solar channels (0.6, 0.8 and 1.6 μm) and 8 thermal infrared channels (3.9, 6.2, 7.3, 8.7, 9.7, 10.8, 12.0 and 13.4 μm) (Taravat et al., 2015). Table 2.1 gives a summary of which properties is detectable by the different channels. This is of great advantage since much of the community already knows how to use the SEVIRI radiance observations. The channels have been chosen based on their ability to detect clouds, water vapor and ozone. More information about what the different channels detect is available in the paper *An introduction to METeosat second generation (MSG)* published in BAMS by Schmetz et al. (2002) .

Spectral band	Central wavelength (μm)	Main Gaseous or Observer or Window
VIS 0.6	0.635	Window
VIS 0.8	0.81	Window
NIR 1.6	1.64	Window
IR 3.9	3.92	Window
WV 6.2	6.25	Water Vapor
WV 7.3	7.35	Water Vapor
IR 8.7	8.7	Window
IR 9.7	9.66	Ozone
IR 10.8	10.8	Window
IR 12.0	12	Window
IR 13.4	13.4	Carbon Dioxide
HRV	0.75	Window and Water Vapor

Table 2.1: *Summary of spectral bands, central wavelength and their respective retrieval abilities (Schmetz et al., 2002).*

Prior to the launch of the METeosat researchers discussed the temporal frequency suitable for observing weather phenomena. The METeosat first generation had a temporal resolution of 30min (Stöckli et al., 2019). For the second generation, 15min intervals were chosen to best cope with the short lifetime and rapid deformation of clouds. It was also suggested that a temporal frequency of 1 to 10min is necessary for tracking cumulus type clouds (Schmetz et al., 2002). This is in agreement with the Table 1.3 in Lohmann, Lüönd, and Mahrt (2016, p. 19) stating the lifetimes of different types of clouds.

A *ring* of geostationary satellites located at equator together provide global coverage (excluding polar regions). The altitude of the satellite determines its forward velocity. To achieve a geosynchronous orbit, a satellite needs to maintain a height of $\sim 36000km$ (Bley and Deneke, 2013). This height lead to a coarser spatial resolution than other satellite products retrieved in other orbits.

The first METeosat Second Generation (MSG-1) was launched 28 August 2002, and became operational January 29, 2004 when it was renamed METeosat-8. As a measure to reduce gaps, the MSG system provides a two-satellite system, one operational and one standby.

The standby satellite scans when the operational satellite experiences technical failures. The operational satellite at nadir points at 0^o latitude. A full disk is 3712×3712 pixels, and is sampled in cycles of 15min. This also include the on-board processing (Schmetz et al., 2002).

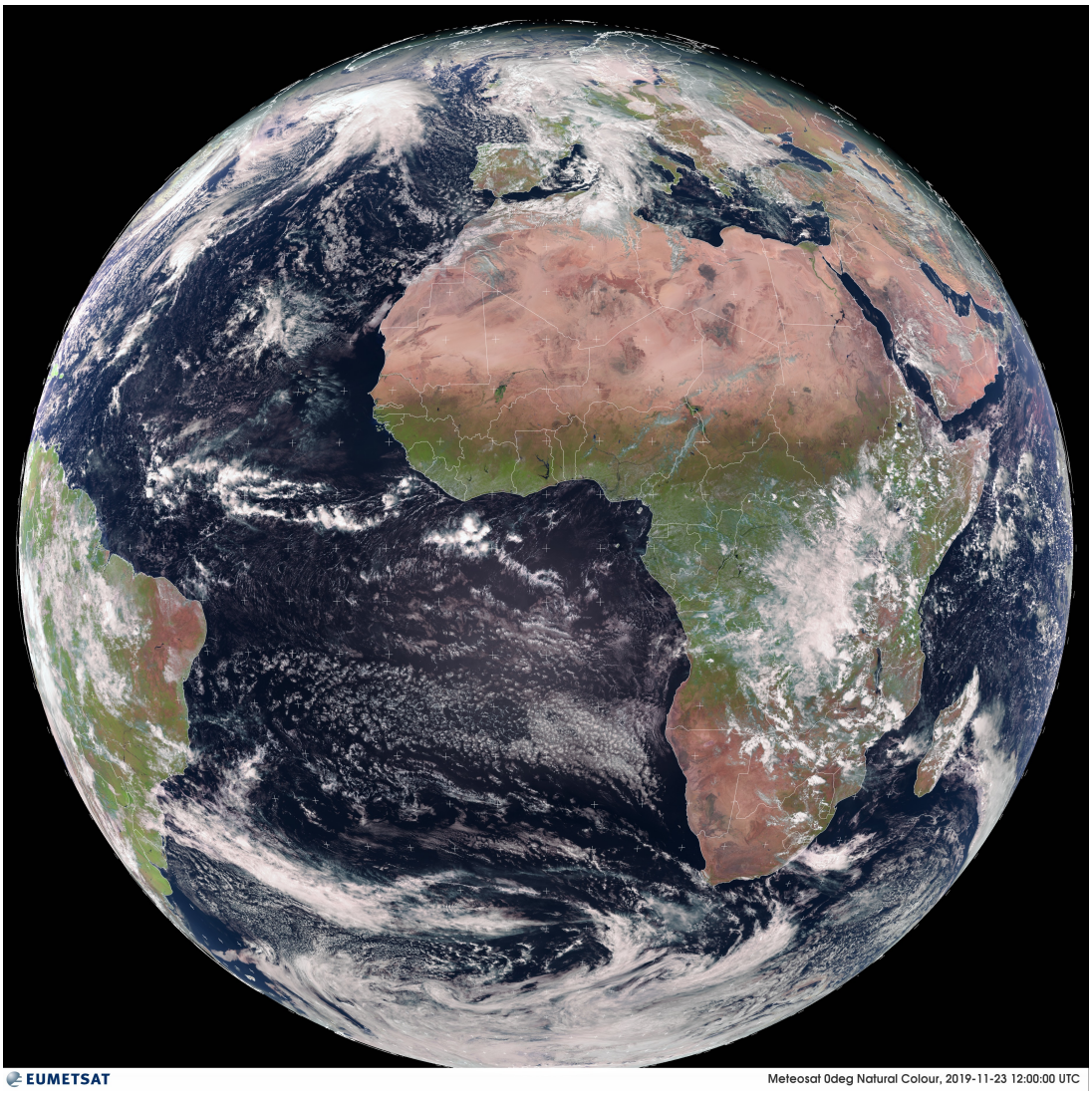


Figure 2.5: *The view of Earth by SEVIRI on MSG, in naturally enhanced colors. This snapshot is dated noon November 23, 2019. By studying this image it becomes clear that clouds are influenced by circulation (EUMETSAT, 2020).*

The operational cloud detection algorithm is carried out pixel by pixel. Post processing involves re-classifying isolated pixels. There is a lot of effort invested in new detection algorithms including spatial structures. One of the methods that show potential here is deep learning (Drönner et al., 2018, Jeppesen et al., 2019).

2.5.4 EUMETSAT Cloud Mask

The EUMETSAT cloud mask (CLM) consist of four classes, described in Table 2.2. These

Class	Description
0	Clear sky over ocean
1	Clear sky over land
2	Cloudy
3	No data/ outer space

Table 2.2: *Description of classes in EUMETSAT Cloud Mask product.*

classes are derived from almost all channels except the broadband High-Resolution Visible Channel (HRV) and isolated pixels are reclassified (Derrien and Gléau, 2005). The cloud mask is distributed in GRIdded Binary or General Regularly-distributed Information in Binary form (GRIB), without coordinates and network Common Data Form (netCDF), processed with coordinates and reshaped into the standard rotation, as displayed in Figure 2.5.

3 Numerical Methods

This section introduces the computational methods used in generating the numerical experiments conducted in this thesis, starting with a short introduction to artificial intelligence. This is followed by a brief introduction to the biological mechanisms the algorithms in this study draw inspiration from. This helps to gain insight into possible applications of different structures. The task of forecasting in time and space requires two types of intelligence. One is computer vision, to understand the spatial relation and use the underlying physical properties. The other is sequential modelling to understand the temporal evolution.

Two approaches will be explored: Autoregressive models (AR) and Convolutional Long Short-Term Memory Network (ConvLSTM). The AR model describes a time varying process, depending linearly on its previous values. The ConvLSTM approach is used to find a non-linear relation that describes phenomena varying in both time and space. The aim of this study is to determine whether the aggregation of linear models, or the more advanced non-linear model are better at predicting the complex cloud phenomena varying in time and space.

3.1 Artificial Intelligence

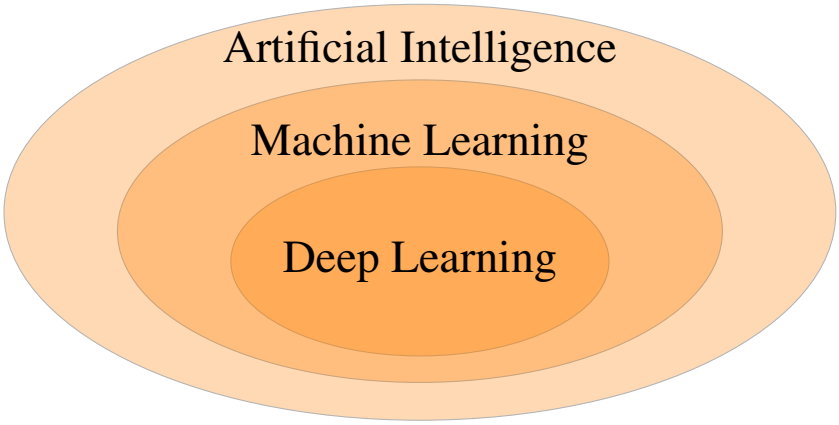


Figure 3.1: *Graph illustrating the subfields of AI. ML is a subfield of AI, DL is again a subfield of ML. The sketch is inspired by Figure 1.1 in Chollet (2017, p.5).*

In encounters with geoscientists the author often gets the question: *What is the difference between machine learning and artificial intelligence?* The short answer is: they are not different fields, but ML is a subfield of AI. Algorithms developing its own "knowledge"

from supplied examples fall into the category of ML and not the base category of AI. ML is distinct in that it attempts to deduce rules and go beyond human intuition using a complex net of interactions.

In fact, it is worth mentioning that there is a subfield of ML known as DL (see graph in Figure 3.1). DL is at the frontier of AI research, with many recent advances being made in this subfield. The popularity of DL can be partly explained by its flexibility. This flexibility allows DL to be applied across many domains. The algorithms discussed here are simply a mathematical framework for learning model representations in data. The process of training is repeated until the network reaches an acceptable performance. In other words, the potential is limited by the data and the effectiveness of the training procedure applied.

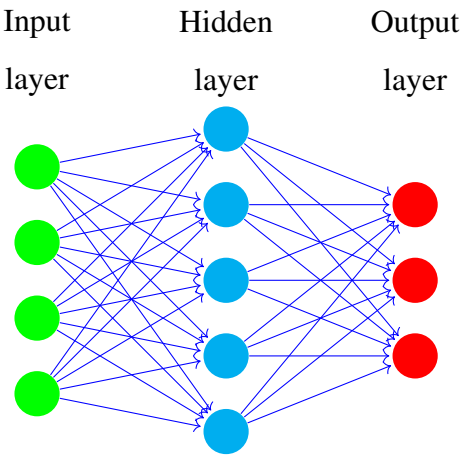


Figure 3.2: *Fully connected feed forward neural network with one hidden layer. The connections between the layers are the weights. The sketch is based on the example provided by Fauske (2006) .*

AI in general and DL in particular emerged from considerations of perception and cognition in biology. Many of the DL network architectures draw inspiration from the human brain. The architecture of DL, while distinct from biological computing, is named such that concepts in neuroscience and computing can be treated analogously. For example using building blocks such as nodes (artificial neurons), weights (connections between nodes), rules of signal propagation, activation (transfer function) and learning algorithms (training algorithms).

Figure 3.2 illustrates a simple artificial neural network. The circles illustrate nodes (neurons). Nodes belonging to the same layer is shown in one colour. Arrows illustrate weights, the

connections between the layers. The figure shows that nodes belonging to the same layer are not connected, but nodes in consecutive layers are connected with weights.

Sequence modelling draws parallels to the human memory. This type of modelling requires information about earlier stages, retained in memory. Simple models have one memory centre. Drawing inspiration from the brain, other more complex models make the distinction between a short term and a long term memory centre. Finishing others' sentences is a simple task for humans. *The clouds are in the ... sky* (Olah, 2015), should not come as a surprise to anyone.

AI started with the idea of automating tasks normally performed by humans. Three factors determine advances in the field of AI: data, hardware, and algorithms (Chollet, 2017). This explains why there often is a significant time gap between an idea and breakthroughs in the architectures and results. Convolution neural networks (CNN), for example, were conceptually developed in the 80s (Fukushima, 1980), but a lack of sufficient computing power (hardware) kept their use in hibernation until 2012, when a CNN (AlexNet) won the ImageNet challenge, an image recognition contest (Krizhevsky, Sutskever, and Geoffrey E. Hinton, 2012).

Geoscientists may be more familiar with the concept of "calibration" when it comes to statistical models, which essentially is the same process. In the context of deep learning, "deep" refers to the number of layers contributing to a network. DL expands the ideas from ML using deeper networks, *i.e,* more layers, enabling networks to capture a more complex relationship between input and output variables.

Figure 3.3 illustrates a deeper version of the network displayed in Figure 3.2. A layer is a set of nodes. The connections between the layers are the trained units, also known as weights.

"Intelligence", in the context of artificial intelligence, is still a topic of debate. Traditionally, a machine would be considered intelligent if it could beat a human at a given task (Chollet, 2019). For computer chess, this was achieved in 1997 when IBM's DeepBlue beat Garry Kasparov. Researchers had learned how to build a "rule-based" chess-playing AI, but not a program that could generalize to anything beyond similar boardgames. In retrospect, scientists have realized that most architectures are not well matched to human intelligence.

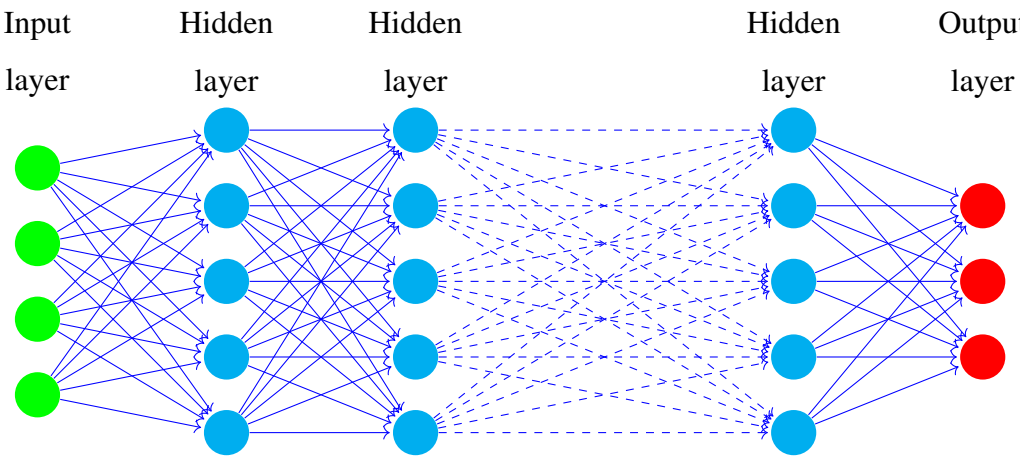


Figure 3.3: *Deep fully connected neural network. An extension of the network presented in Figure 3.2. The sketch is based on the example by Fauske (2006) .*

There are several different types of machine learning, each suitable for solving different tasks. Figure 3.4 shows the types of ML and their subcategories. These subcategories also exist for deep learning, the only difference being the number of layers used.

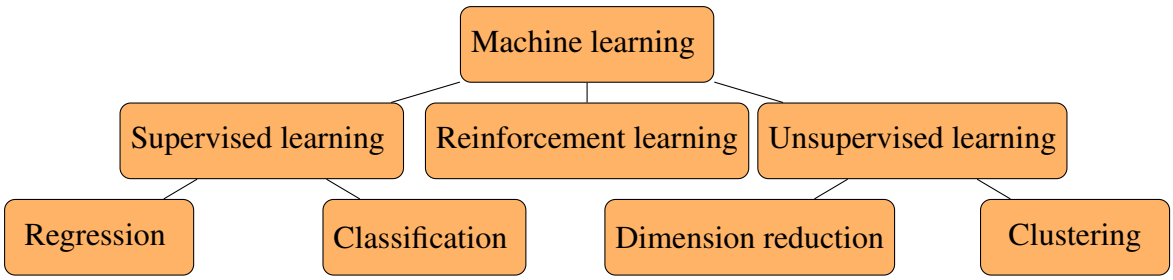


Figure 3.4: *The graph shows the different type of machine learning and their subcategories.*

- **Supervised learning**: part of machine learning concerned with learning the relation between input data, x and labeled data, y.
  - Regression

    predict continuous values. Replicating a function.
  - Classification

    discrete, since it assigns a category to the input.
- **Unsupervised learning**: Detecting patterns in unlabeled data.
  - Clustering

    Grouping a set of data points into a predescribed number of groups

  - Dimension reduction

    Reducing the number of random variables under consideration.

- **Reinforcement learning**: Goal oriented algorithms.

3.2 Artificial Neural Networks

Because of the interdisciplinary nature of this thesis, this section provides a thorough walk through the relevant algorithms using the computational graphs, and relevant equations.

Artificial neural networks (ANN) are composed of nodes (artificial neurons) and weights. Returning to Figure 3.2, it illustrates nodes as circles and weights as arrows. It is an example of a 2-layer ANN. The nodes are structured in layers, illustrated using different colors. The input layer contains four input nodes, the hidden layer five nodes, and the output layer three nodes. The dimensions of the input and output layers are determined by the task at hand. The number of hidden layers and the number of nodes are tunable parameters, called hyper-parameteres. Nodes of one layer are only connected to the the nodes of the following layer. Weights are the relative strength of the connections between nodes in neighboring layers. Large networks of these simple neurons are able to perform complex calculations.

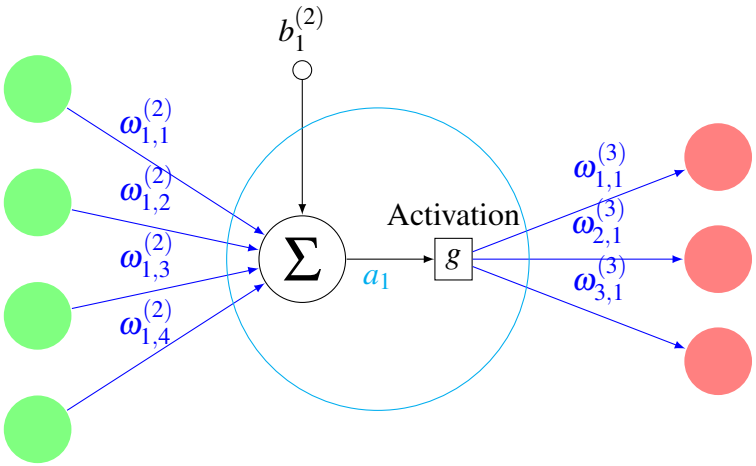


Figure 3.5: *Computational graph showing the components participating in the activation of a neuron in the hidden layer. This example shows a 2-layer neural network with four input nodes and three output nodes. The number of nodes in the hidden layer doesn't affect the activation, because they are not connected. The sum of the weighted inputs and the bias is passed to the activation function, inside the hidden layers, producing the activation of the neuron. This is again passed to the output neurons. Modified sketch based on example from stackexchange (2019) .*

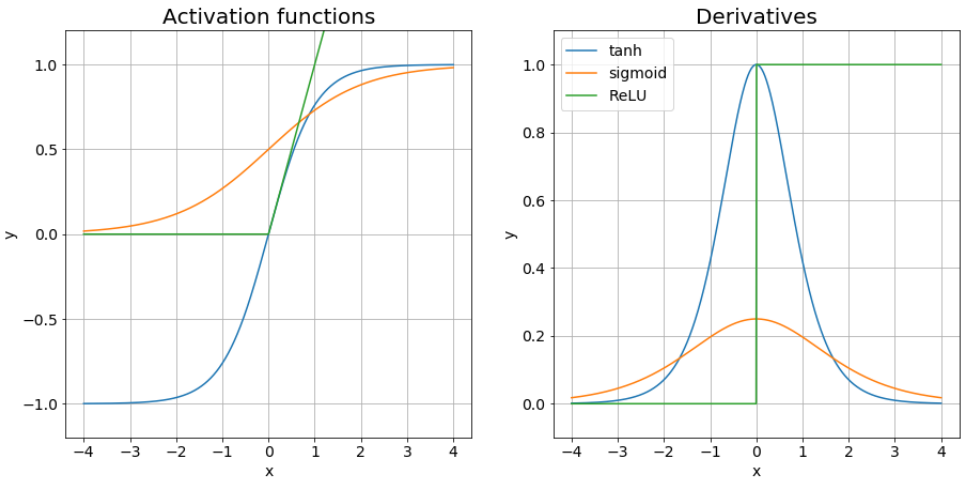


Figure 3.6: *Activation functions and their derivatives.*

Figure 3.5 shows the computation which takes place in a node in the hidden layer, focusing on a circle in the middle column in Figure 3.2. The sum of the weighted inputs and bias are sent trough the activation function, g, producing the activation. This function, g, is a hyperparameter, set before the training starts. Popular choices are rectified linear unit (ReLU), Sigmoid function (σ) or hyperbolic tangent (tanh), their graphs are shown in Figure 3.6 and their mathematical expressions in Equations (3.1), (3.2) and (3.3) respectively. It is called the sigmoid function, named after the greek letter sigma, shaped like an S.

$$ReLU\,(x) = \begin{cases} \text{x}, & \text{if x} \geq 0 \\ 0, & \text{else} \end{cases} \tag{3.1}$$

$$\sigma\,(x) = \frac{1}{1+e^{-x}} \tag{3.2}$$

$$tanh\,(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} = \frac{e^{2x}-1}{e^{2x}+1} \tag{3.3}$$

Equation (3.4) describes the activation of a node in a arbitrary layer, L. b_L denotes the bias, w_L is the weights matrix. and n_L is the number of nodes. g_L denotes activation function in a arbitrary layer.

$$\mathbf{a}_L = g_L\left(\sum_{i=1}^{n_L} \mathbf{W}_{L,i}\mathbf{x}_i + b_L\right) \tag{3.4}$$

Repeating the procedure for the next layer, $L+1$, the activation, a_L from the previous layer is weighted and passed trough the activation function, g_{L+1}, generating the activation, a_{L+1}.

In a forward pass, this is repeated for all layers, until the output layer is reached. Adapting Equation (3.4) to the example in Figure 3.4 can be done by inserting $L = 1$ and $n_L = 4$.

The choice of activation function, g, in the output layer is task specific. Regression problems use linear activation. Classification problems need functions able to discriminate between the number of classes.

Backpropagation is the fundamental mechanism used in "teaching" neural networks, as first appreciated in its full importance in 1986 when it was published by Rumelhart, G. E. Hinton, and Williams. The trick is to use the performance metric as a feedback signal to adjust the weights in the direction of the lowest loss score for the current example. A training instance is passed trough the network producing an output (red node). The network's output error is computed as the difference between the produced output and the corresponding targets. For each consecutive layer, moving in reverse from the output to the input, the contribution from every connection between adjacent layers is computed. For a more mathematical description of the backpropagation algorithm see Nielsen (2015) .

3.2.1 Convolutional neural networks

Computer vision is a field of artificial intelligence concerned with interpreting the visual world. One popular structure for visual tasks is the Convolutional Neural Network (CNN). Computers see images as a grid of numbers, often decoded in red, green and blue (RGB)

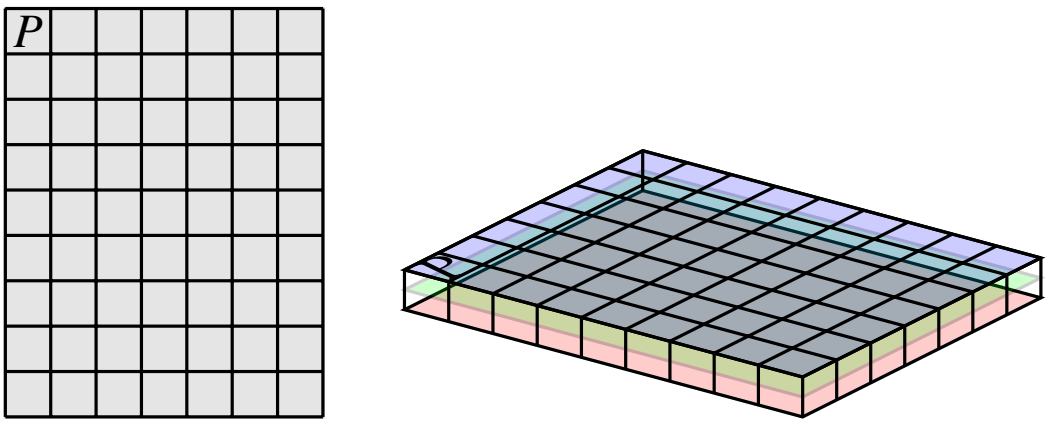


Figure 3.7: *Decoding a image into Red-Green-Blue (RGB) channels involves transforming a 2-dimensional image into a 3-dimensional tensor. Inspired by Figure 1 in Shi et al. (2015)*

.

channels. Figure 3.7 shows the transformation of a two-dimensional image to a 3-dimensional

tensor. The "P" shows the connection between one pixel ("picture element") and a volume. Each of the grid cells (pixels) contains the signal from the colour decoded into values ranging from 0 to 255. The machine needs to learn how to extract the necessary information about these pixels to perform a task. More layers increase the model's ability to extract these complex structures, resulting in improved model performance. Figure 3.8 shows the mathe-

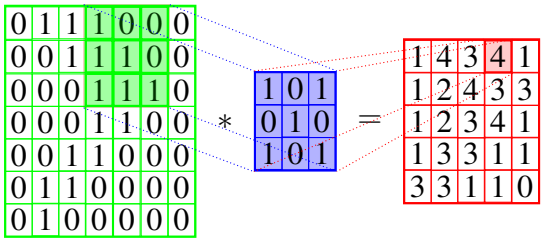


Figure 3.8: *Diagram showing a convolutional operation. Modified sketch based on stackexchange (2019) .*

matical operation convolution as the sum over element-wise multiplication of the filter and input. The filter is blue, this is placed over the filled green section, producing the red output pixel. The entire red grid is called a feature map (output map). The green grid is the input, overlaid with blue indicating the pixels contributing to the activation, of the red pixel. In Figure 3.8 this would be the value 4. *Receptive field* is known as the pixels contributing to the activation in a pixel (i.e. the value) (Luo et al., 2016). For instance the receptive field of the shaded red pixel is the shaded green submatrix.

Convolving a filter over the input image generates a feature map, a 2-dimensional activation. If it happens to be the last layer, it is common to refer to the results as the output instead, although this is merely a difference in terminology. Figure 3.9 shows a 2D-convolution with filter of size 3×3. Filters are often square (not a strict requirement), and the height f_h and width f_w are odd numbers. The origin is the position of the filter which is above the current output pixels. The connections between the layers are intended to illustrate the part contributing to the pixel, as well as highlighting the receptive field. In order to include the outermost pixels, the input area is padded with zeros around the edges (grey boarder).

Working with RGB images requires 3-dimensional convolution; since the dimensions of the input determines the dimensions of the convolution, it is commonly referred to as simply convolution. As mentioned earlier, neural networks are structured as a stack of layers. Each layer is again a stack of channels or feature maps. The output from the previous layer be-

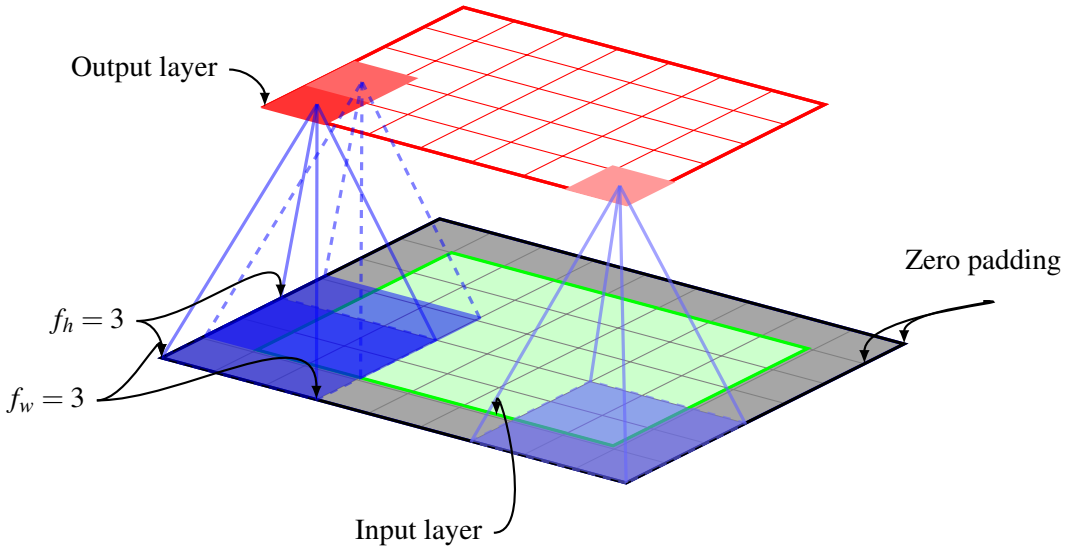


Figure 3.9: *The sketch shows the connections between input and output layer. This example uses a* 3×3*-filter (blue), zero-padding (grey) resulting in equal dimensions for input (green) and output (red). The different colours illustrate the connections between input and output pixels. The input pixels contributing to the output, is called the receptive field. The zero padding is added to keep the input shape. Inspired by Figure (13-3) in Gron (2017) .*

comes the input to the next layer. Feature maps, activations, and outputs are all the result of a convolution, produced at different points within a neural net. The activations are computed based on an input volume, including information across channels. A 3-dimensional convolution collapses information on multiple colors into a single value.

Figure 3.10 shows a two layer convolutional neural network trained on RGB-images. The input layer is an RGB-image. The first convolutional layer has seven channels (feature maps), these are produced by seven filters. Filters are trained to extract useful features. The second convolutional layer is produced by five filters, all convolving layer 1. This is a simplified network, made shallow for for the purpose of illustration; a functional CNN would require many layers to extract useful information from an image.

Given raw input (*i.e,* normalized images), the first layers detect low level features like edges, corners and circles. Later layers assemble the features to more complex structures like houses or dogs. The dashed volumes represent the receptive field for different pixels, illustrated as circles. Since each of the layers depend on the previous one, the receptive field of a node, "P" depends on a large portion of the input image. Small filters allow you to focus on small features in the data, while larger filters allow you to identify coarser relations.

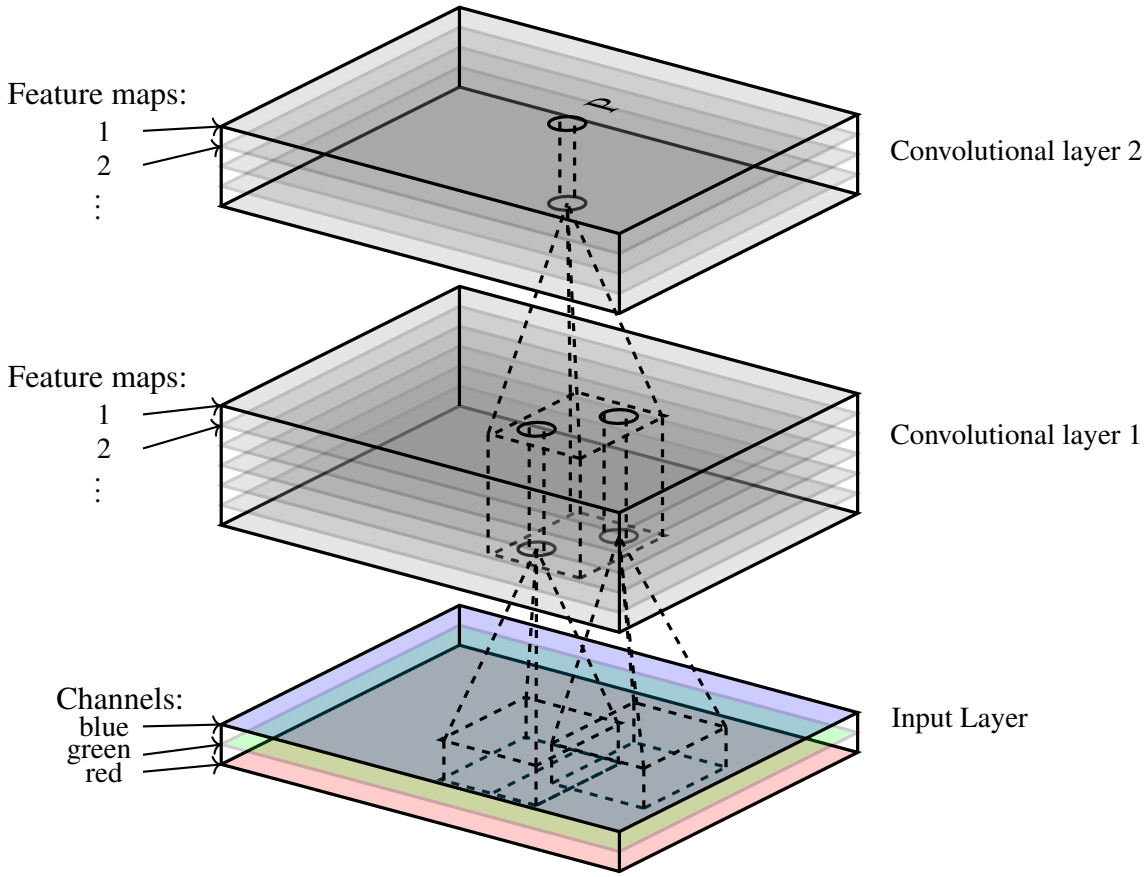


Figure 3.10: *First two layers of a convolutional neural network trained on RGB-images. Each convolutional layer contains multiple filters, thus producing stack of feature maps. Each layer learn the representation of the previous layer. The trailing layer get this stack as input, producing activations based on all channels. For each layer it contains representations of the structures found in the previous layer. The filters are the weights trained to find useful structures. In each convolutional layer multiple of these filters are passed over the image. The dashed volumes illustrate the receptive fields of a pixel, "P". The receptive field of a node in the second layer is larger than the one in first, since a pixel inherent the receptive fields of the nodes in its receptive field. Inspired by Figure (13-6) in Gron (2017)*

.

Unlike the fully connected neural network layers (see Figure 3.2), the nodes in the output layer are not connected to all the input nodes, only the nodes within their receptive fields. The filters contain the trained units. Its dimensions determine the size of the feature it can detect. One filter convolves the entire image, searching for a single feature. When it finds this particular feature it activates, propagating this signal into the feature maps.

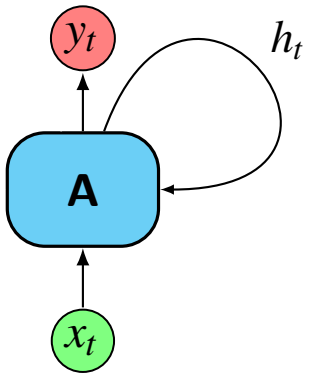


Figure 3.11: *Simple one layer recurrent network, x_t denotes the input element of the training sequence, h_t denotes the hidden state, y_t denotes the output of the neuron and A denotes a artificial recurrent unit. In the simplest cases without output activations the hidden state and the output is identical. Inspired by Olah (2015) .*

3.2.2 Recurrent Nets

A Recurrent Neural Network (RNN) is a class of artificial neural networks developed for studying patterns in sequential data such as time series, audio, or text. Figure 3.11 shows the structure of a simple RNN. In very simplified terms, the recurrent unit, A, receives an input, x_t, produces an output, y_t and passes hidden state, h_t back to itself. The hidden state contains the information about what you have learned so far. The output at each time step is dependent on the previous inputs.

Figure 3.12 shows the recurrent network unrolled in time. This way of structuring it resembles the earlier structures like ANN (see Figure 3.2). The connection between the nodes are a directed graph along a temporal sequence. The hidden state from the previous step is fed into the next, here the recurrent units, A can be considered a copy. For each time step it is fed with new examples, h_0 is only dependent on x_0, while h_t is dependent on the entire training sequence $x_0, x_1, \cdots, x_t$. This example shows a one layer recurrent network. All time steps are passed through the same node, updating the connections between the input, output and hidden states. In this way, the RNN reuses the weights for all time steps, performing the same task on all inputs along the sequence. This reduces the complexity of parameters and in turn lowers the risk of overfitting, obtaining a more general relation between input and output.

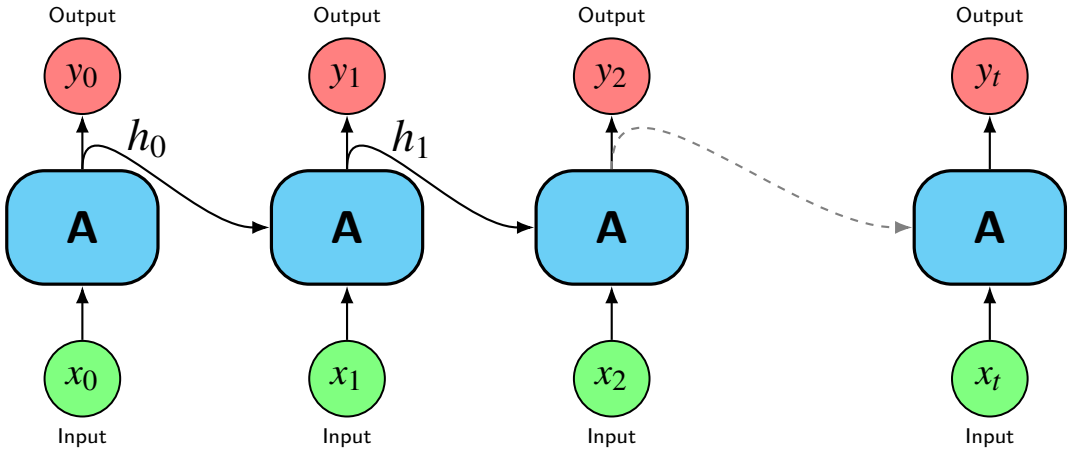


Figure 3.12: *Unrolling Figure 3.11 in time, t denotes the length of the training sequence, the elements x_t are the input, y_t the output and h_t is the hidden states, rpresenting the memory. Inspired by Olah (2015) .*

Learning long term dependencies can be challenging and is done by backpropagating the error signal thought the network. Working with longer sequences, the error signal tends to approach zero or infinity. Exploding gradients can cause the weights to oscillate. Learning from small or vanishing gradients takes a very long time, and might not produce a useful outcome at all.

3.2.3 Long Short-Term Memory Network

In order to address challenges in predicting long sequences presented in the previous section, Hochreiter and Jürgen Schmidhuber created the Long Short-Term Memory (LSTM). Their design, documented in a paper from 1997 (Hochreiter and Jürgen Schmidhuber, 1997), outperformed previous memory networks by regulating the flow of information provided to a recurrent network at each time step. They propose a new method for learning in order to alleviate the issues with exploding or vanishing gradients. This approach is called constant error flow.

The original memory cell contains input and output gates. A gate is a structure that can be opened or closed. Having values ranging from 0 to 1, it truncates the noise signal from the input and the output. Gers, J. Schmidhuber, and Cummins suggested in 1999 to add an additional gate, the forget gate. The idea was to enable the LSTM to reset parts of its own memory. Resetting releases internal resources and enables even more learning. In very simplified terms, the forget gate learns which part of the cell state, the long-term memory, it should forget. The input gate learns the information from the input it should add to the cell

state. The output gate learns which information it should pass to the output. The assembled memory unit is displayed in Figure 3.13a. The information flow thought the cell is regulated by three gates; the forget gate (see Figure 3.13c), the input gate (see Figure 3.13d) and the output gate (see Figure 3.13f). These gates are neural networks.

The long-term memory is shown in Figure 3.13b and the short term memory is referred to as the hidden state. The cell state is affected by some linear interactions, this is very simple, thus the flow often remains unchanged. The hidden state is the output passed to the next cell. Structures like gates regulate the flow of information.

The gates are neural networks layers with a particular activation function. The sigmoid-function discussed in Section 3.2 truncates the output from the gates to range between zero and one. Recall, Figure 3.6 shows the sigmoid and tanh functions. Zero represents a closed gate, while one describes a fully open gate. At each time step in the sequence the LSTM receives input x_t and the previous hidden state, h_{t-1}. These are passed trough all gates.

Making a prediction requires the forward propagation of information through the network. Figure 3.13c shows that based on the new input and the previous hidden state, the forget gate determines which instances from the memory to remove. Regulating the information that stays in memory frees up space, allowing the network to learn new things. The weight of the gate, W_f, is initialized to 1, thus it cannot forget anything until it has learned to forget. Exhibiting the same behavior as the original LSTM units.

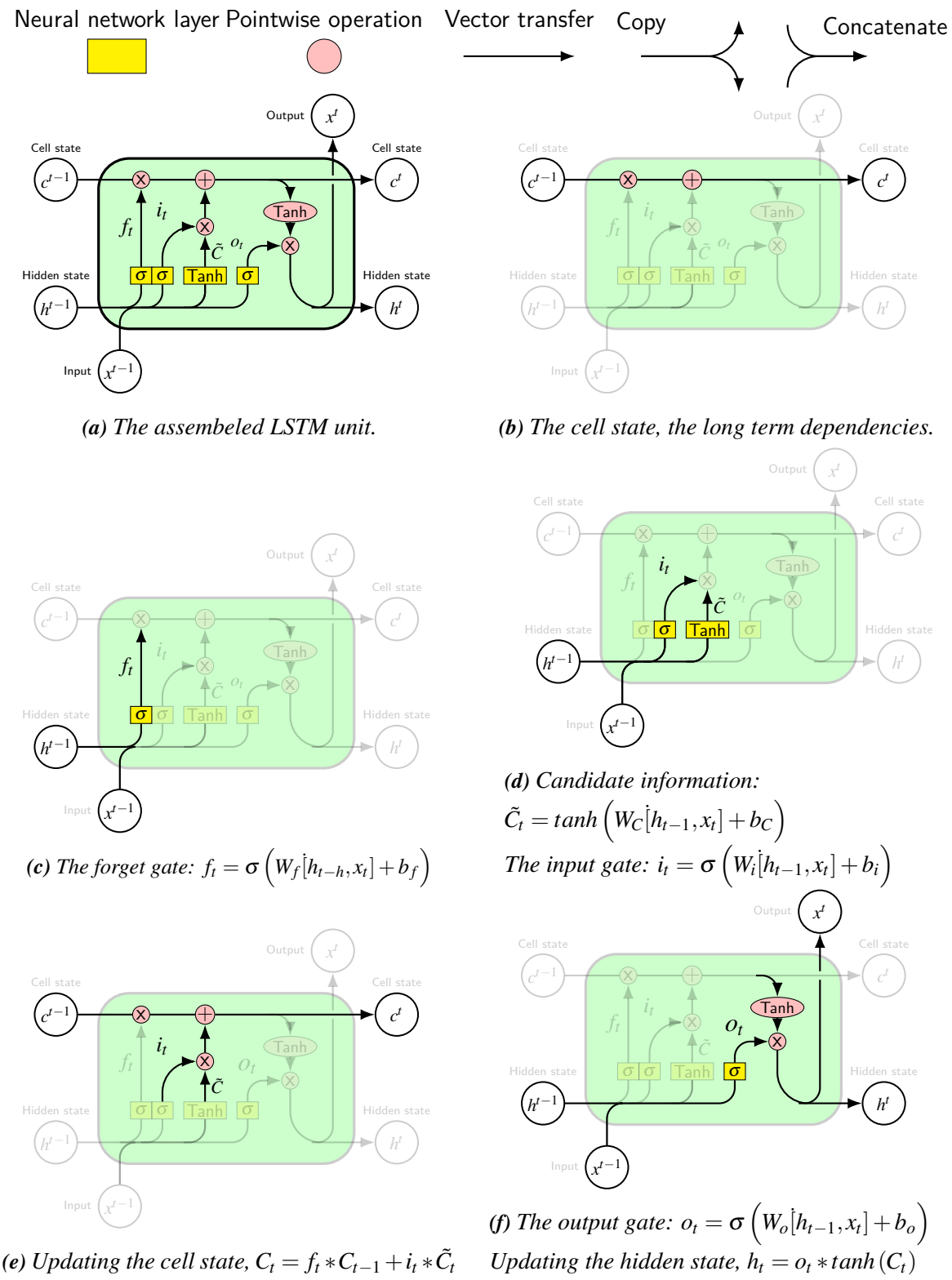
Figure 3.13d shows two processes, one determining the candidate information based on the input and another the computations in the input gate. The candidate information is filtered by the multiplicative input gate. This determines what information to add to the cell state.

Figure 3.13e shows how the cell state is updated. Using multiplicative gates, first the forget gate removes information. Then the output from the input gate adds the useful information from the input to memory. The input gate regulates what information to store from the input. The aim of the input gate is to clean the input by reducing the noise signal. These computations are also shown in Figure 3.13d.

Figure 3.13f illustrates how the output gate gets updated. Passing the cell state through the function $tanh$, it is given a value in the range from -1 to 1. The output gate determines what to remove from the cell state and pass as the hidden state. This gate aims to remove noise

from the output, preventing misrepresentations of the hidden state (short-term memory). This forces the hidden state to always take values in the range minus one to one (Hochreiter and Jürgen Schmidhuber, 1997). In summary, at each time step some memories are removed and others are added.

The process of training is repeated until the network reaches an acceptable performance. The principle of learning is the same as for the simpler architectures. However, the mathematical description of the backpropagation algorithm gets more complicated. The reverse pass measures the error gradients over all connections traversing backward to the input layer. Like before, based on the error signal the weights are adjusted in the direction of the lowest error.



(a) *The assembeled LSTM unit.*

(b) *The cell state, the long term dependencies.*

(c) *The forget gate:* $f_t = \sigma\left(W_f[h_{t-h}, x_t] + b_f\right)$

(d) *Candidate information:*

$\tilde{C}_t = tanh\left(W_C[h_{t-1}, x_t] + b_C\right)$

The input gate: $i_t = \sigma\left(W_i[h_{t-1}, x_t] + b_i\right)$

(e) *Updating the cell state,* $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$

(f) *The output gate:* $o_t = \sigma\left(W_o[h_{t-1}, x_t] + b_o\right)$

Updating the hidden state, $h_t = o_t * tanh\,(C_t)$

Figure 3.13: *Walk trough the components of a LSTM and the relevant equations. Extension of example provided by Leon (2018) , inspired by Olah (2015) .*

3.2.4 Convolutional LSTM

A variant of the LSTM network is the Convolutional Long Short-Term Memory (ConvLSTM), which was originally developed for the application of precipitation nowcasting in 2015. This architecture has the potential to solve problems varying in time and space. The difference between this and the general LSTM is that the standard ANN (see Figure 3.2) is replaced by CNN (see Figure 3.9). This allows the LSTM network to support multi-dimensional data, capturing its spatiotemporal structure.

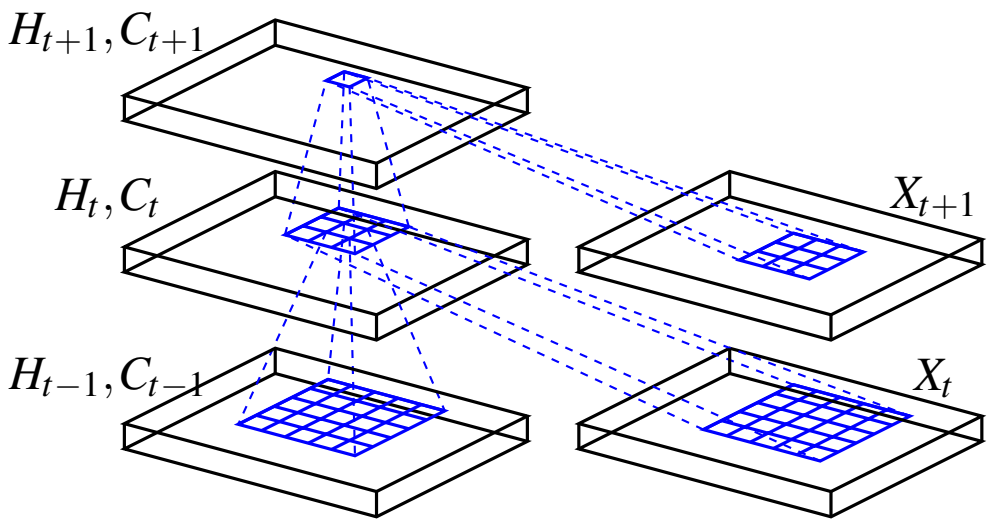


Figure 3.14: *Structure inside ConvLSTM. Illustrating the differences in input to a ConvLSTM cell and a LSTM cell. The inputs and states are now vectors standing on a grid. The actuvation, also know as future states is now a function of the input and passed states of the pixel and its neighbors. This is archived using the convolutional operation in the input-to-state transition and the state-to-state transition. Inspired by Shi et al. (2015) .*

Preserving the structure from Figure 3.13a, and making small changes to the equations used to forward propagate the input, the multiplicative gates are now replaced with convolution. Figure 3.14 shows the dimensions of states and input, the inner structure, in a ConvLSTM.

Equations (3.5) - (3.9) describes the forward propagation through a ConvLSTM.

$$f_t = \sigma\left(W_{xf} * x_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f\right) \tag{3.5}$$

$$i_t = \sigma\left(W_{xi} * x_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i\right) \tag{3.6}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ tanh\left(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c\right) \tag{3.7}$$

$$o_t = \sigma\left(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o\right) \tag{3.8}$$

$$H_t = o_t \circ tanh\left(C_t\right) \tag{3.9}$$

Here $\circ$ denoted the Hademand product, which is a component wise multiplication, and * is convolution. For an arbitrary time step, t, H_t denotes the hidden state and C_t is the cell state, σ is the sigmoid function (see Equation (3.2)), *tanh* the hyperbolic tangent (see Equation (3.3)), and $W_{\text{component,gate}}$ denotes the trained weights of components and gates (Shi et al., 2015).

3.2.5 Padding

As mentioned related to Figure 3.9, the convolution operation shrinks the dimensions of the feature map, according to Equation (3.10). The degree of shrinking depends on the filter size, $f \times f$, padding, p and stride, s. Stride determine how you convolve around the input volume. It is the step length between applied filters. For an input of dimensions $n \times n$, the resulting output dimension, $o \times o$ is given by Equation (3.10).

$$o = \frac{n + 2p - f}{s} + 1 \tag{3.10}$$

Padding zeros along the edges has an additional benefit of including the signal originating at the boundaries. For some applications it is useful to have the same shape of input and output. This is called “padding same”. Equation (3.11) calculates the amount of padding necessary to preserve the dimension of the input volume. Derived from Equation (3.10) inserting $o = n$ and solving for p.

$$p = \frac{n(s-1) - s + f}{2} \tag{3.11}$$

3.3 Autoregressive Models

In the following discussion, these conventions are adopted. Typographical emphasis is used to distinguish between dimensions, y_n denotes a discrete value at position or time, n, the

vector, $\mathbf{y}$ and $\mathbf{Y}$ denotes the matrix. $\mathbf{X}$ always contains the input data, and $\mathbf{y}$ always contains the predicted. The markers$\tilde{}$,$\hat{}$and$\bar{}$can be used in combination with the above, describing the $\tilde{\mathbf{y}}$ denotes the desired value, the predicted $\hat{\mathbf{y}}$ and the mean of $\mathbf{y}$, $\bar{\mathbf{y}}$.

The AR-model is a form of linear model where values from previous time steps are included as predictor variables. In discrete form it is described by Equation (3.12) and in matrix form by Equation (3.13), here a prediction at time, n, $\hat{y}_n$ is the linear combination of the values at i, earlier time steps, y_{n-i}, their corresponding weights, β_i, and β_0, the bias (intercept), corresponding to the intersection of a function on the y-axis.

$$\hat{y}_n = \beta_0 + \sum_{i=1}^{N} y_{n-i}\beta_i \tag{3.12}$$

$$\hat{\mathbf{y}} = \mathbf{Y}^T\beta \tag{3.13}$$

Expanding the traditional AR-model to include other predictors, x_i, yields the expression,

$$\hat{y}_n = \beta_0 + \sum_{i=1}^{N} y_{n-i}\beta_i + \sum_{j=1}^{p} x_j\beta_{j+p} \tag{3.14}$$

Here p denotes the number of predictors, and β_{j+p} the corresponding weights. The other symbols are described above referring to Equation (3.12).

The weight indices might seem unnecessary complicated at first, but ease the transition to the vector equation, $\beta = [\beta_0, \beta_1, \ldots, \beta_i, \beta_{i+1}, \ldots, \beta_{p+i}]^T$. The corresponding input matrix, $\mathbf{X} = [1, y_{n-1}, \ldots, y_{n-i}, x_0, \ldots, x_p]$. This can be formulated as a least-squares problem, and the analytical solution to this optimization problem is found by minimizing the Mean Squared Error (MSE) loss,

$$L(\tilde{y}, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}(\tilde{y}_i - \hat{y}_i)^2 \tag{3.15}$$

where $\hat{y} = \mathbf{X}^T\beta$. Equation (3.17) describes the optimal solution β. Inserting Equation (3.13) into (3.15), solving Equation (3.16) with the respect of β yields the solution presented in Equation (3.17).

$$\frac{dL}{dX} = 0 \tag{3.16}$$

$$\beta = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}\tilde{\mathbf{y}} \tag{3.17}$$

The optimal solution is the best solution based on the training data available. The analytical solution is computationally very fast, as long as the matrix $\mathbf{X}^T\mathbf{X}$ is non-singular and thus its

inverse exits. For the problem considered in this study there are far more observations than the number of parameters. This is a indication that $\mathbf{X}^T\mathbf{X}$ most likely will be invertible.

3.3.1 Evaluation

Model evaluation is done based on metrics. This metric is used to determine the ability of a model to perform a certain task. Varying in space and time, the metrics used to evaluate the models in this study need to be 3-dimensional.

These metrics are set based on criteria measuring properties of the algorithm, for instance mean squared error measures the squared distance to the true value, giving a idea on how well it fits to the target. The expression is shown in Equation (3.18), notice that it is not scaled. MSE is non-linear and numbers in the range between 0 and 1 shrink, while others grow exponentially. Mean Absolute Error (MAE) is the absolute value between the same distances, see Equation (3.19).

$$MSE(\hat{Y},\hat{\tilde{Y}}) = \frac{1}{n \cdot m \cdot k} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \sum_{k=0}^{t-1} (y_{i,j,k} - \tilde{y}_{i,j,k})^2 \tag{3.18}$$

$$MAE(\hat{Y},\hat{\tilde{Y}}) = \frac{1}{n \cdot m \cdot k} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \sum_{k=0}^{t-1} \left| y_{i,j,k} - \tilde{y}_{i,j,k} \right| \tag{3.19}$$

The absolute value or sum of squares is to avoid penalizing points on the lower side of the line, it has an additional benefit of not having to deal with negative distances.

3.4 Related work

Deep learning is a young field and ConvLSTM an even younger model. Few studies have been conducted within DL applied to sequence modeling. The task of object detection or classification has to this date received the most attention.

The ConvLSTM architecture was developed by Shi et al. (2015) for the task of precipitation nowcasting. The model used gridded data and the spatial dimension is 100×100 after resizing it using filters to remove noise. The dataset, named Radar Echo, has 8148 training, 2037 validation and 2037 test samples. Each sample consists of 20 frames, 5 used for input and 15 used for prediction, in other words based on 5 frames the model predicts 15. Shi et al. report that they have managed to training a model, named $ConvLstm3 \times 3 - 64 - 3 \times 3 - 64$. This model has two layers, both with 64 hidden states, used in combination with 3×3 filter. Theses results are compared to a FC-LSTM-2000-2000 and a traditional model for precipitation nowcasting, the ROVER algorithmn. Both LSTM versions optimize the crossentropy

error of 15 prediction samples. This metric aims to predict the most likely event. The ConvLSTM-model showed superior performance over LSTM and ROVER.

Sun and Liu (2018) applied it to the air quality forecasting problem, using high resolution data from a period up to two years. They use a combination of distributed data. Weather data is provided in grids, and air quality observations are point measurements. The weather data volume at each timestep is a tensor of shape $(21 \times 31 \times 5)$. Sixteen months of hourly data were divided into sequences of 72 frames, 24 for input and 48 for prediction. They use a learning rate scheduler dropping from 0.01 to 0.001 after five epochs, and gradient clipping to avoid exploding gradients when training longer sequences. This is combined with the following filters; 1×1, 3×3 and 5×5. Two or three layers with either 256 or 128 hidden states. The model names are given following this structure, described using an example. *ConvLSTM 3x3-256-2*, is a ConvLSTM model using a 3×3 filter, in combination with 2 layers, both of 256 hidden states. Their work showed a significant improvement over LSTM.

4 Results and Discussion

4.1 Dataset - European Cloud Cover

This section presents the developed algorithms necessary for the compilation of the dataset European Cloud Cover (ECC). It is pieced together from two sources, reanalysis, ERA5 and METeosat Second Generation (MSG) cloud mask. The common period of 2004 to 2018 is selected and downloaded size is 17Gb, stored in netCDF-files.

Reanalysis products is as close one can get to observations continuous in time and space. There is multiple global reanalysis datasets available and they are all different. It depends on the forecast assimilation system used and observations assimilated (Fujiwara et al., 2017). ERA5 was elected because of it's fine resolution and resent release date of January 2020.

Several candidate satellites were considered before arriving at the combination of datasets presented in this chapter. Spatiotemporal consistency and resolution were given top priority. The variable cloud mask is provided by many satellites, bringing valuable information in itself, but also for the retrieval of other variables restricted to cloud free conditions, such as humidity. The satellite product chosen for this project is the METeosat Second Generation (MSG). This satellite is in geostationary orbit, and has an exceptional temporal resolution, with scans every 15min. Knowing that the average lifetime of a cloud is 60min or less, the selected data set was found to be the most suitable for the purpose of this study (Lohmann, Lüönd, and Mahrt, 2016, p. 19). The finished dataset, described in detail below, is named European Cloud Cover (ECC).

4.1.1 Domain

The geographical domain has, for this project, been restricted to latitude, $\theta \in [30, 50]$ and longitude $\phi \in [-15, 25]$. The resulting dimensions of the grid become 81×161 pixels. Figure 4.1 shows the domain included in this dataset. The domain covers Central Europe and North Africa. 54% of the domain is land, and 46% is ocean or sea.

4.1.2 Physical basis of variable decision

The overall goal is to investigate whether basic meteorological variables such as temperature, pressure and humidity are sufficient for prediction of cloud fractional cover with a reasonable

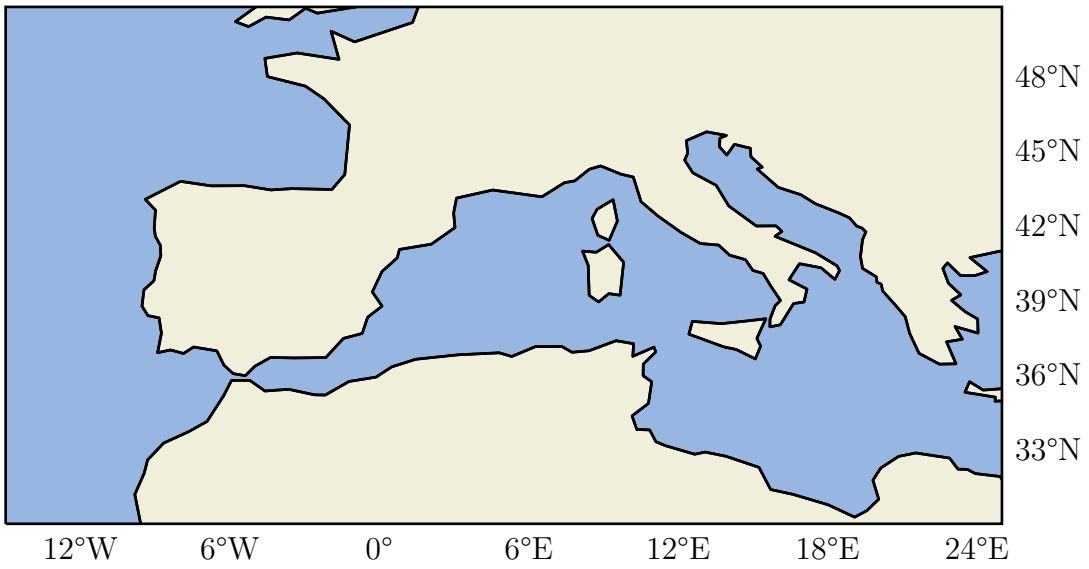


Figure 4.1: *Map showing the domain in the projection available in ECC.*

accuracy. Cloud dynamics is far more complicated than what can be describes by these variables, as discussed in Section 2.1.

This project is a *proof of concept study* aimed to demonstrate the feasibility of using AI to parameterize CFC. Employing data driven learning to represent cloud physics and dynamics in its full complexity requires measurements from technologies not yet invented. If they did exist, the computational cost would be large and there is no guarantees of state-of-the-art performance. However, large amounts of data is collected on a daily basis. Understanding how to utilize the information in these measurements despite its flaws (gaps and artifacts) is of major importance.

Precipitation formation and cloud optical thickness are affected by changes on a microphysical level. However, they are undeniably closely related to the macrophysical properties of the cloud as well. Imagine precipitation without a cloud fractional cover. Reliable estimates of large-scale variables are available from using reanalyses or other climate models. Restricting the focus here to the macrophysical aspects of clouds makes it reasonable to choose these as explanatory variables, thereby ensuring that it is possible to build usable application of this parameterization in the future.

For this dataset, all these large-scale variables are produced by ERA5. Some variables have been chosen because they are reliable and fundamental meteorological variables, e.g. temperature and surface pressure. Others were chosen because they are essential in cloud for-

mation, e.g. specific and relative humidity. The variables were all retrieved from the surface or the closest pressure level (1000hPa). The rest of this section gives a brief introduction to their role in cloud formation.

From the weather maps on the news, low and high pressure systems might be familiar terms. Low pressure systems are often associated with precipitation, while high pressure systems are associated with nice weather. The Earth is not equally heated due to the its spherical geometry. Warmer air rises, generating a low pressure at the surface. As the air rises the temperature within the rising air parcel decreases. From Equation (2.2) it can be shown that colder air can retain less vapor, enhancing the rate at which saturation is achieved. Under supersaturated conditions some of the vapor condenses, generating cloud water, forming a cloud, and occasionally precipitation. Low latitudes and the mid-latitudes of the summer hemisphere are often associated with this type of convective motions, generating cumulus type clouds.

In areas of lower pressure, the surrounding air will flow toward the low pressure center to offset the pressure difference. Induced by Earth's rotation the Coriolis effect forces winds of low pressure systems to swirl counterclockwise north of the equator and clockwise south of the equator, causing an accumulation of air in the center of low pressure system, pushing it to higher altitudes in the atmosphere. A high pressure system exhibits the exact opposite behavior, it swirls in the reverse direction, and the air flows from the center. Diverging air masses cause sinking motions of parcels from higher in the atmosphere to fill the void.

The large scale circulation generate zonal bands of sinking and rising air masses. Warm air rises at the equator as a result of the Earths' spherical geometry, this air move poleward, cools and sinks. The Saharan desert is located in an area of sinking air masses. Despite its warm temperatures, there is little to no cloud formation and associated precipitation in this region.

In zones of rising air, bands of clouds are concentrated around fronts. A front is a transition zone between two types of air masses, usually of different temperatures. Large cloud decks are generated from the lifting of entire layers. This is likely to happen when the air is forced to higher altitudes by fonts (warm and cold) or orography (Lohmann, Lüönd, and Mahrt, 2016, p. 79). Norway lies within the zone of westerlies. Low pressure systems frequently appear along the coast, resulting in a cloudy and wet region.

The dataset includes both relative and specific humidity. Relative humidity is a measure of how much vapor the air contains, relative to how much it can hold at a certain temperature. At relative humidity of 1 the air is at its dew point, and for higher values clouds form. These conditions are commonly referred to as supersaturated, consequently the variable relative humidity is "limited" to the range from 0 to 1. However the variable is still a measure of the identical ratio, and this explains why values exceeding 1 is present in the dataset. Relative humidity is unitless and for higher values the air is more humid.

Specific humidity is the ratio between mass of vapor and mass of air, with unit of $kgkg^{-1}$ (Lohmann, Lüönd, and Mahrt, 2016, pp. 53-54). Whether relative or specific humidity is the better predictor is not clear a priori. The data is gathered from the model level closest to the surface, at an altitude of 1000hPa. For large scale motions, when friction and the centrifugal force are less important, the wind field can be derived from the pressure gradients, this is know as geostrophical winds (Lohmann, Lüönd, and Mahrt, 2016, pp. 81-84).

Observed weather systems depend upon local and large scale interaction between variables. Different combinations are thought to explain different cloud conditions. For instance surface pressure is thought be a good predictor of frontal systems. The surface humidity, on the other hand is not. Fronts appear at high altitudes in the atmosphere that surface humidity is not thought to take part in saturating these airmasses. Therefore, humidity is expected to have a low predictability for such an event.

In shallow (and deep) convection systems, the air rises from the surface and the temperature and surface humidity is though to explain this phenomena well. Rising airmasses will in turn generate low pressure at the surface. The variables temperature, humidity and pressure are assumed to be reasonable predictors for convective activity and the cumulus clouds responsible for afternoon showers.

4.1.3 Area Weighting Regridding Scheme (AWRS)

Computing cloud fractions based on the cloud mask requires a regridding scheme. Common schemes for solving similar tasks are mean, nearest neighbor or area weighting. For this particular task, the pixels are of uneven size and the area weighted scheme seemed most appropriate.

This section provides a step-by-step description of the necessary data processing done for the compilation of ECC, transforming clouds masks provided in *space-view* to cloud fractions on

a uniform grid. EUMETSAT doesn't provide suitable software to tackle this particular task (personal communication EUMETSAT staff). Building the dataset requires the implementations of software with functionality to perform the regridding, described in detail below, named the Area Weighting Regridding Scheme (AWRS).

The regridding algorithm consists of two modules; *(1) detection* and *(2) area weighting algorithm*. Let the subscripts denote the dataset pertaining to a particular grid. Grid_{MSG} refers to the space-view grid of the METeosat Second Generation and grid_{ECC} refers to the uniform grid originating from ERA5. Note that grid_{ECC} is identical to $\text{grid}_{\text{ERA5}}$. The *detection algorithm*, computes the co-location of pixels across grid and determines the contributing pixel from grid_{MSG} to $\text{grid}_{\text{ERA5}}$.

The pixels are then classified into the different categories such as *corner*, *center*, *left*, *right*, *upper* and *lower boundary*. These categories are later used to isolate the portion of the pixel contributing to grid_{ECC}. The second module, *the area weighting*, consist of the category based area weighting algorithm based on the developed equations.

Equations describing grid area

Since Earth is spherical the pixels areas are computed using spherical coordinates. Figure 4.2 shows a square projected on to a sphere. Deriving the equation for computing the area of a square in spherical coordinates, requires integrating over changes in latitude, $d\theta$ and longitude, $d\phi$. In Equation (4.1) the variables of integration are given a prime to keep them distinct from the integration boundaries. The general expression for the area of a square in spherical coordinates, is given by the following integral,

$$A = R^2 \int_{\theta-\delta\theta}^{\theta+\delta\theta} \int_{\phi-\delta\phi}^{\phi+\delta\phi} cos\left(\theta'\right) d\phi' d\theta' \tag{4.1}$$

the equation can be rewritten into,

$$A\left(\theta,\phi,\delta\theta,\delta\phi\right) = 2R^2\left(sin\left(\theta+\delta\theta\right) - sin\left(\theta-\delta\theta\right)\right)\delta\phi \tag{4.2}$$

where $R = 6378km$ denotes the distance to Earth's center, θ the latitude and ϕ the longitude. Equations (4.1) and (4.2) can be related to Figure 4.2 by setting $d\theta = 2\delta\theta$ and $d\phi = 2\delta\phi$. The implementation is scaled by R, to avoid unnecessary large values. Multiplication followed by division of the same number provides no additional information, and there is the off chance of introducing additional numerical errors.

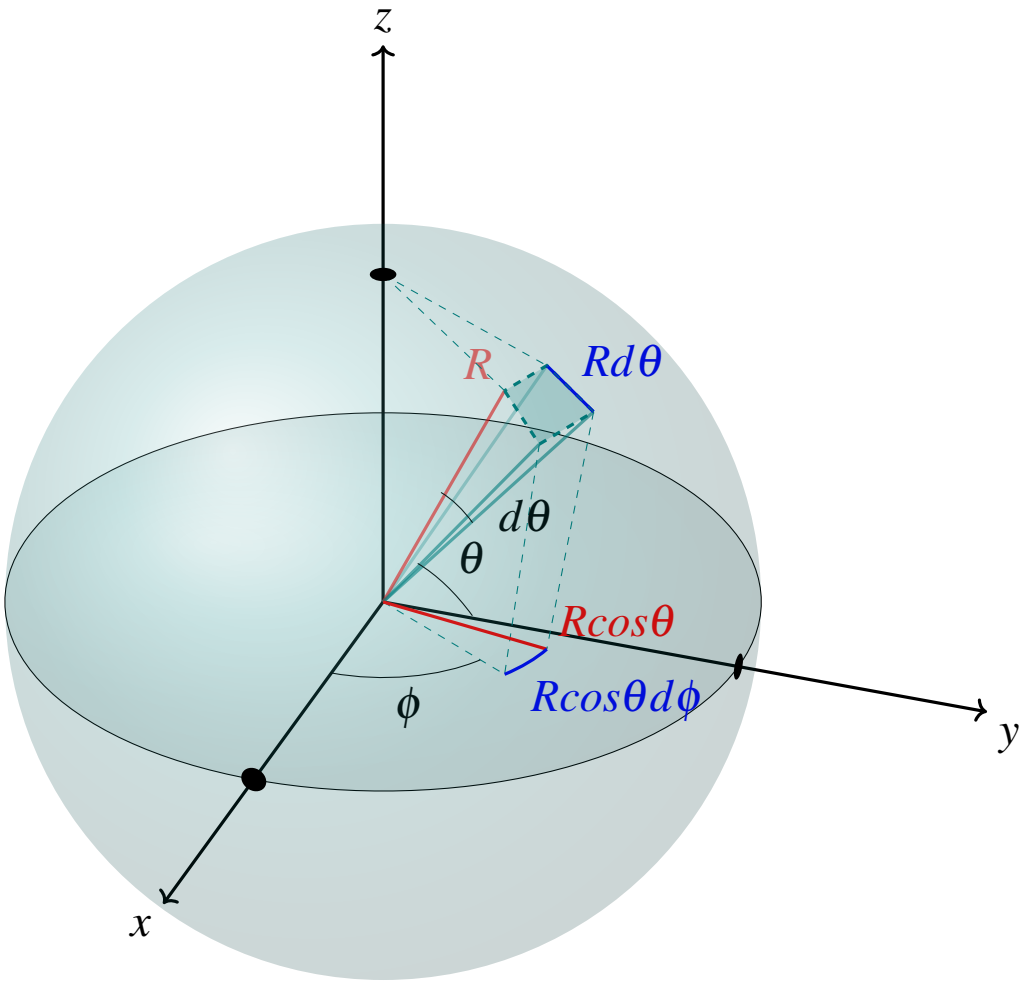


Figure 4.2: *Mathematical properties of a square projected onto a sphere.*

The equations used to compute the area weighted cloud fractional cover are as follows,

$$CFC_{ECC} = \frac{1}{A}\sum_{i=0}^{N} a_i m_i \tag{4.3}$$

where,

$$A = \sum_{i=0}^{N} a_i \tag{4.4}$$

Inserting $m_i = 1 \; \forall \; i \in [0,N]$ into Equation (4.3) results in $CFC_{ECC} = 1$, independent of a_i, proving that the minor overlap between pixels in Figure 4.4 does not affect the range of cloud fractions, which remains between 0 and 1.

Estimating properties of grid$_{\text{MSG}}$

The coordinate information is provided in grids of latitudes, θ (degrees north), and longitudes, ϕ (degrees east) values. The coordinate represents the center of a pixel. To acquire the information about the extent of cells in a non-uniform grid requires some simplifications.

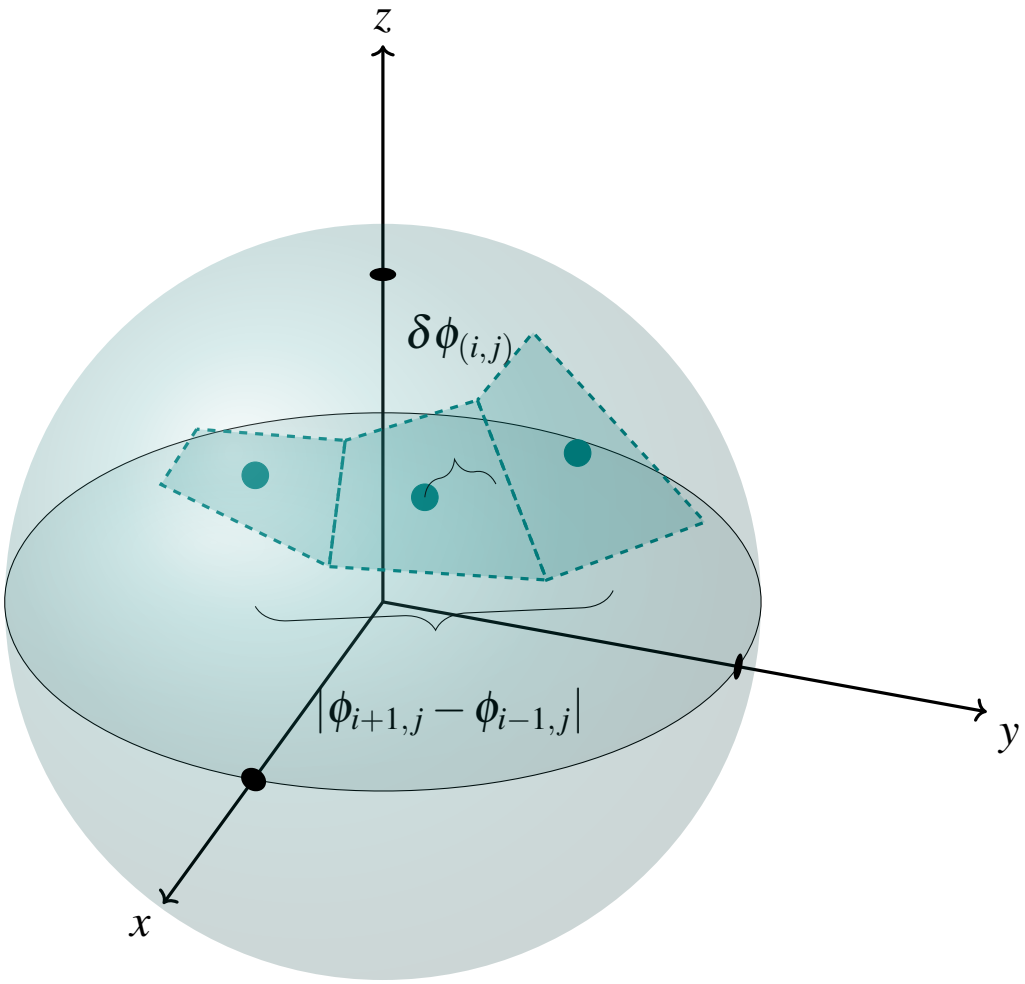


Figure 4.3: *Illustrating the relative size of three neighbouring pixels on a sphere in the space-view grid provided by EUMETSAT. The following example explain the changes in longitude, ϕ, varying in eastward direction, denoted i. The expression is shown in Equation* (4.5)*. The same principles applies in the latitudinal direction, see Equation* (4.6)*.*

Computing the area weighted average of cloud masks requires detecting the pixels of grid_{MSG} contributing to the $\text{grid}_{\text{ERA5}}$ cell, and computing their area. The pixels are classified into five categories; *center*, *left*, *right*, *up* and *down boundary*. The cloud mask pixels are smallest at the nadir point, increasing in all directions. Figure 4.3 illustrates this by showing an example of three neighboring pixels. In this example the apparent pixel size increases going eastward. Consequently, there is an asymmetry between neighboring pixels. The distance from the center to the right and left neighbor is unequal. In order to take advantage of the analytical expression derived in Equation (4.2), the extent of a contributing pixel is approximated by averaging the right and left distances. The categorization is illustrated in Figure 4.4 using different colors.

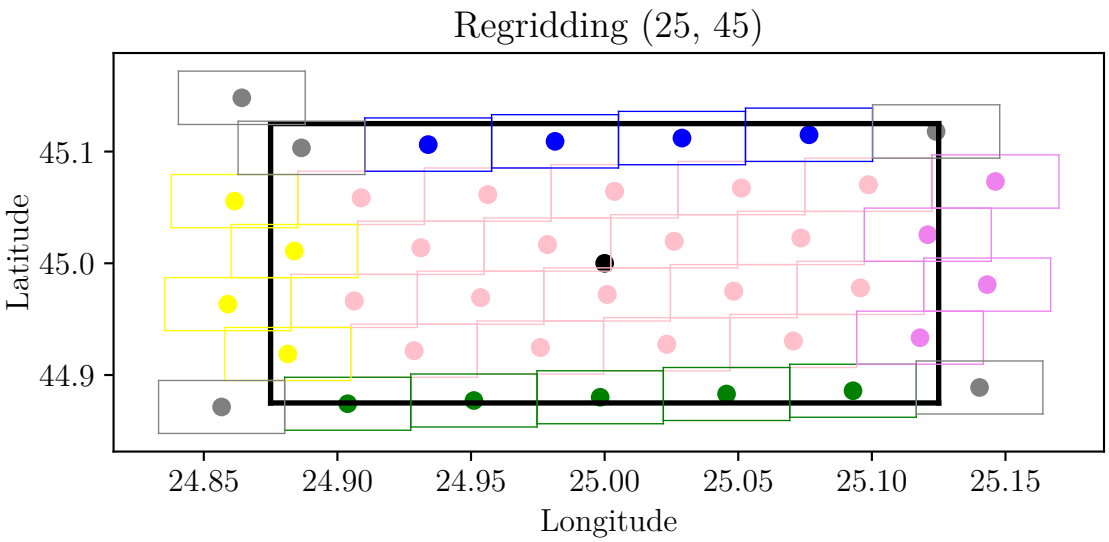


Figure 4.4: *Example showing the contributing pixels to the remapping of pixel* (25,45). *The pixels from the satellite are classified into corner (grey), center (pink), right (purple), left (yellow), lower (green) and upper (blue) boundary. The dense black line is the pixel in* $grid_{ECC}$*, and the other pixels shows the contributing pixels from* $grid_{MSG}$.

Approximations of $d\phi$ and $d\theta$ have been made based on the two-dimensional fields of latitude and longitude values according to the below equations. Longitudinal values vary along the first dimension, denoted with index i. The estimated half of the extent of a pixel in grid_{MSG} is approximated by Equation (4.5). Latitudinal changes are in north-south direction, along the second-axis, here denoted with index j. Equation (4.6) describes the distance from the center to the upper and lower boundary.

$$\delta\phi_{i,j} = \left| \frac{\phi_{i+1,j} - \phi_{i-1,j}}{4} \right| \tag{4.5}$$

$$\delta\theta_{i,j} = \left| \frac{\theta_{i,j+1} - \theta_{i,j-1}}{4} \right| \tag{4.6}$$

The "square" in grid_{MSG} resembles a trapezium, as illustrated in Figure 4.3. As far as the author knows, an analytical solution for the area of a trapezium in spherical coordinates does not exist. Also, it is not clear if approximating the area with another numerical method would reduce the overall uncertainty. However, it would most likely increase the computational time. A visual inspection of the relation between both grids is provided in Figure 4.4. Based on the small overlapping areas, it appears to be a reasonable simplification at the latitudes and longitudes of interest. This particular pixel was chosen since the largest overlap is expected in

the periphery of grid_{ECC}. The inclusion of the boundary pixels involve estimating the "new" center and the extent of the contributing portion, defined as the subset falling within the boundaries of grid_{ECC}. Due to the small overlap between contributing pixels, occasionally more than four pixels are classified as corners. For the sake of simplicity the corner pixels where omitted from the calculations of cloud fraction. This may have caused the circular pattern shown in Figure 4.5. The areas decrease poleward, as illustrated in Figure 4.6. This again has most likely enlarged the radius of the circular pattern.

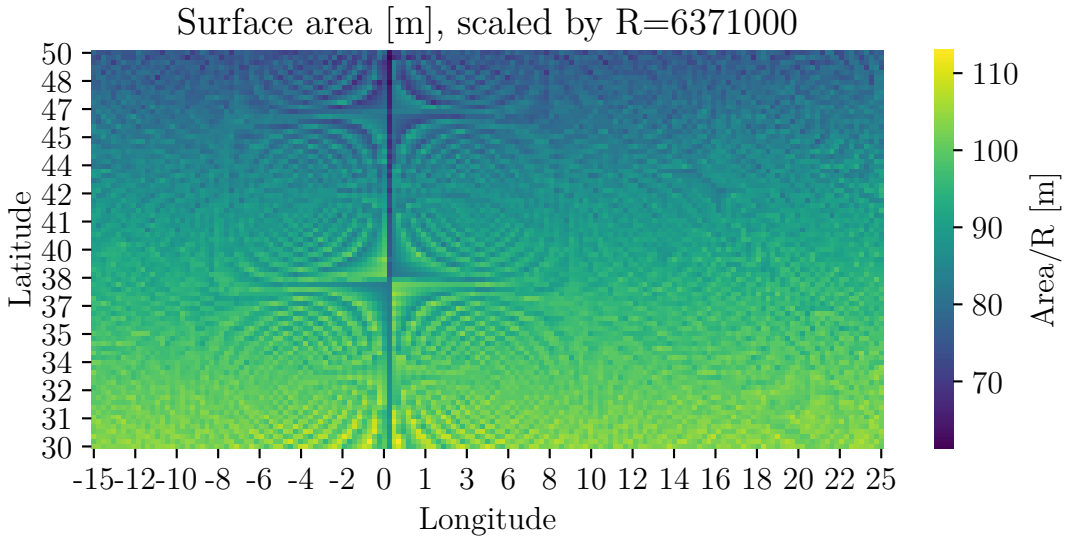


Figure 4.5: *The surface areas of pixels decreasing poleward. Note, the pattern appears to be more pronounced close to the meridian decreasing in both east and west direction.*

4.1.4 Verification of AWRS

The regridding scheme is a set of algorithms. Before the production of the full dataset it is important to verify the correctness of the algorithms of AWRS.

The computations of the areas are the foundation of the AWRS, and it is crucial to verify that the algorithm is correct. CDO (Schulzweida, 2019) provides functionality to compute grid areas of a uniform grid using the code inserted below.

```
$ cdo gridarea era5data.nc gridarea.nc
```

To verify the implementations of Equation (4.2), the grid areas of ERA5 are computed using both CDO and the self implemented algorithm. Both versions produce the same results. Note that the self implemented code is scaled by R.

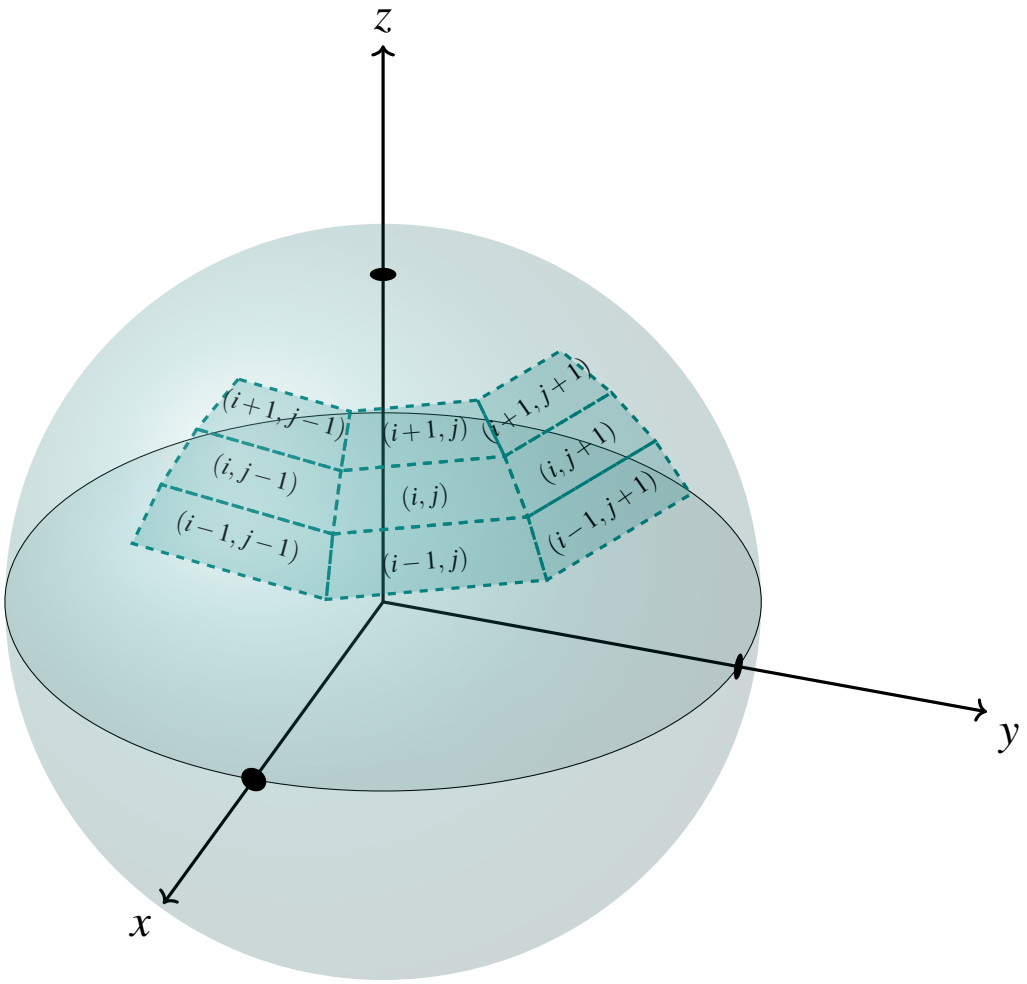


Figure 4.6: *Sketch illustrating the relative size of neighboring pixels in uniform grid of ECC, projected onto spherical coordinates. The areas of pixels in a uniform grid decrease poleward.*

A reason for confusion in the process of developing the AWRS algorithm arose from the different rotations of the data provided by the GRIB and netCDF files. The coordinate information is available in netCDF, but the file-size is too large to store all data in this format, therefore the data is downloaded in GRIB, as mentioned in Section 2.5.4. The GRIB-file is provided as the left and right flipped version of the netCDF-file. A fast modular test to check the regridding routine is to insert the raw data including the land and sea masks. Figure 4.7 shows a example of the regridding of the raw data from May 2, 2009. Clouds are illustrated in white, land mask in teal and sea in purple. The original data also include the category off-earth disk, this is not a member of the chosen subset. By regridding the raw data it quickly becomes apparent whether the correct domain has been used, and evident if the algorithm generates any discontinuities in space.

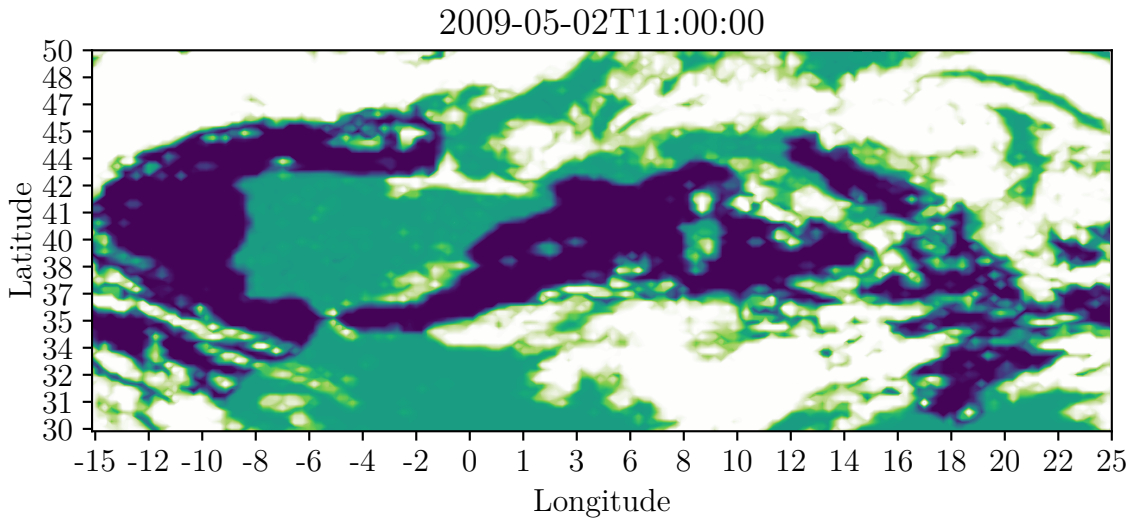


Figure 4.7: *Result from regridding the raw MSG CLM product, which includes land and sea masks in the absence of clouds. Land is illustrated in teal, sea in purple and clouds in white.*

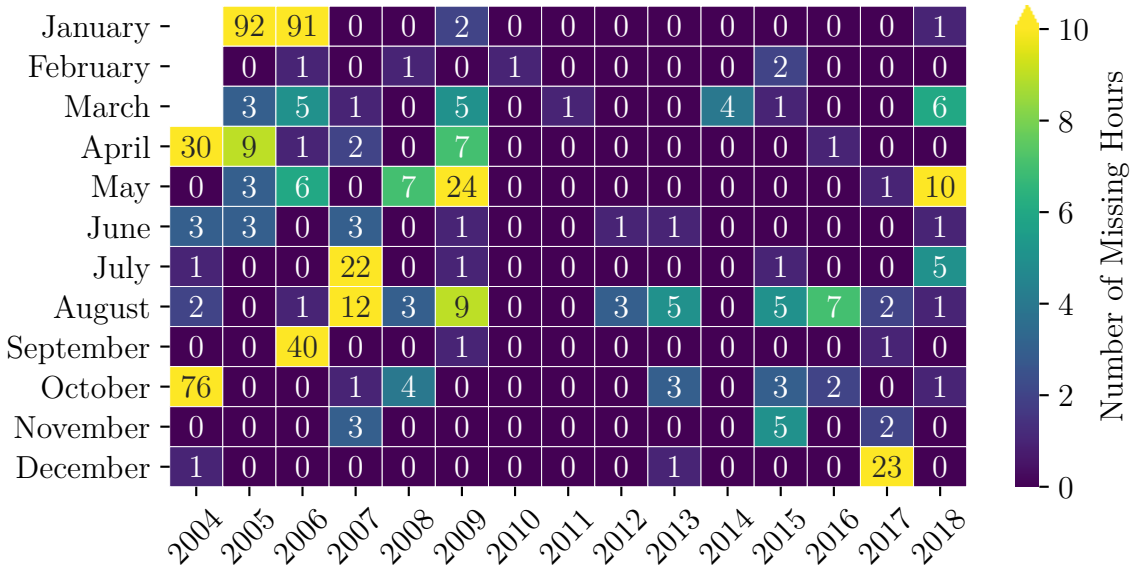


Figure 4.8: *Heatmap summarising missing hours per month for all years.*

4.1.5 Missing Data

Missing values are inevitable when working with observational data. Sensors occasionally fail to collect measurements and data is missing. This can either be individual pixels or entire disks. In this project single NaN values are no pressing issue since they are remapped to fractions by using the area weighting of the other values. Contributing NaN pixels are counted and stored for future use in ECC. In some cases the sensor fails to scan and in other

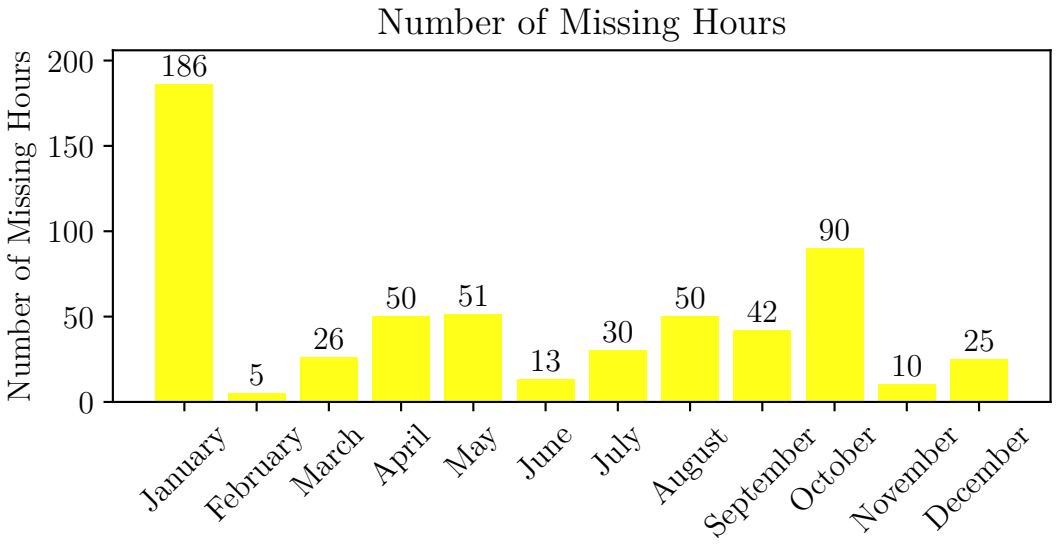


Figure 4.9: *Barplot showing the monthly sum of missing values. This excludes the contribution from the period of 2004 before the satellite was operational.*

cases the data has been destroyed prior to archiving. Independent of the cause, the result is the same, the data can never be recovered. Further down the chain of supply, the missing data can cause issues for the process of downloading data. Corrupt data can enter infinity loops without being detected by EUMETSAT. One request amount to approximately 3.5 months of hourly data. The combination of corrupt data and maximum number of pending requests (20) have cause some delays in preparing the dataset.

Missing timesteps result in missing disks. When available the closest time step within the previous and trailing 45 minutes is used to fill the gap. A summary of missing values per month in the data set is provided in Figure 4.8. Aggregation of missing values per month is presented in Figure 4.9. The plot is ment to illustrate any seasonal biases. The months of 2004 prior to the time at which the satellite became operational are not included in the statistics of missing values.

METeosat provides a two satellite system, and occasionally both the standby and the operational sensors scan at the same time, as mentioned in Section 2.5.3. In cases of technical failures, the standby scan is used. The scans are done from a different nominal position. However, the coordinate systems remain the same, since the standby scan is rectified to the position of the operational satellite before the product is released (personal communication EUMETSAT staff). Comparing simultaneous measurements for the operational and the

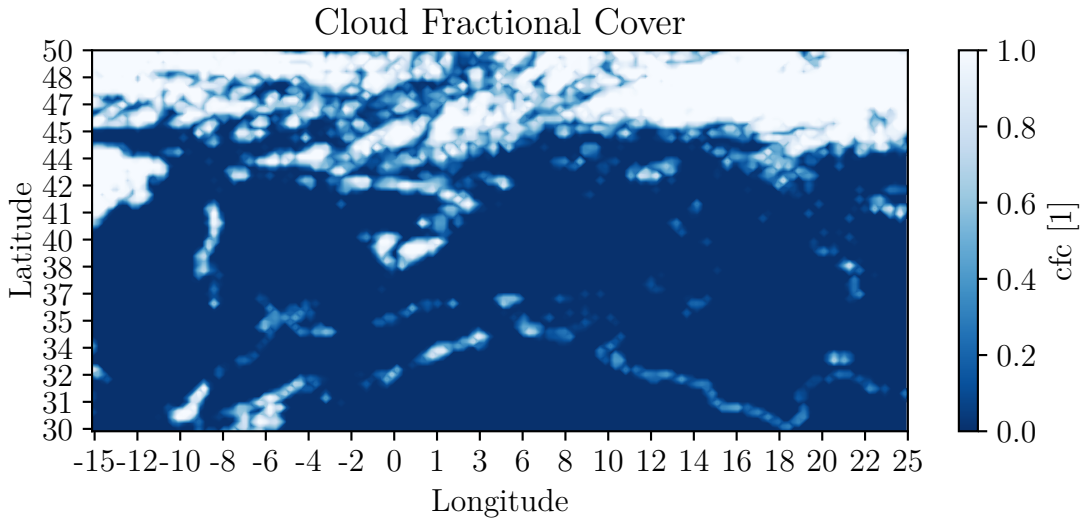


Figure 4.10: *The results after regridding reveal an artifact. This is a snapshot from May 2, 2004 at noon. The frequency of occurrence is currently unknown.*

standby METeosat satellites, it becomes clear that they are dissimilar. However, this does not occur very frequently and there has been no effort in quantifying the magnitude of the parallax, to correct for the bias this may introduce.

Manually generated datasets are prone to human error, especially in the case where users need to download individual time steps to fill the gaps. The instances of missing values were double checked. In summary the workflow has been as follows: the author downloaded the data, detected missing times and manually chose the closest time step available within the previous and trailing 45 minutes. In retrospect, the downloading options (API or GUI) provided by the satellite service should be taken into account when choosing the data.

4.1.6 Masks

The land sea masks are provided by Norwegian Meteorological Institute (MetNo). In its original format the masks have a 0.1^o resolution and global coverage, not including the polar regions. These are regridded to a suitable resolution of 0.25^o using functionality available in PyAEROCOM (MET Norway, 2020). This is a python toolbox developed within the Aerosols Comparisons between Observations and Models (AEROCOM) project. To avoid storing redundant data, only domain specific data is made available in the project supplementary repository. For more details on supplementary material, see Section 4.2.1.

Filters available in the supplementary material are *land*, *sea*, *coastline* and *artifact*. The

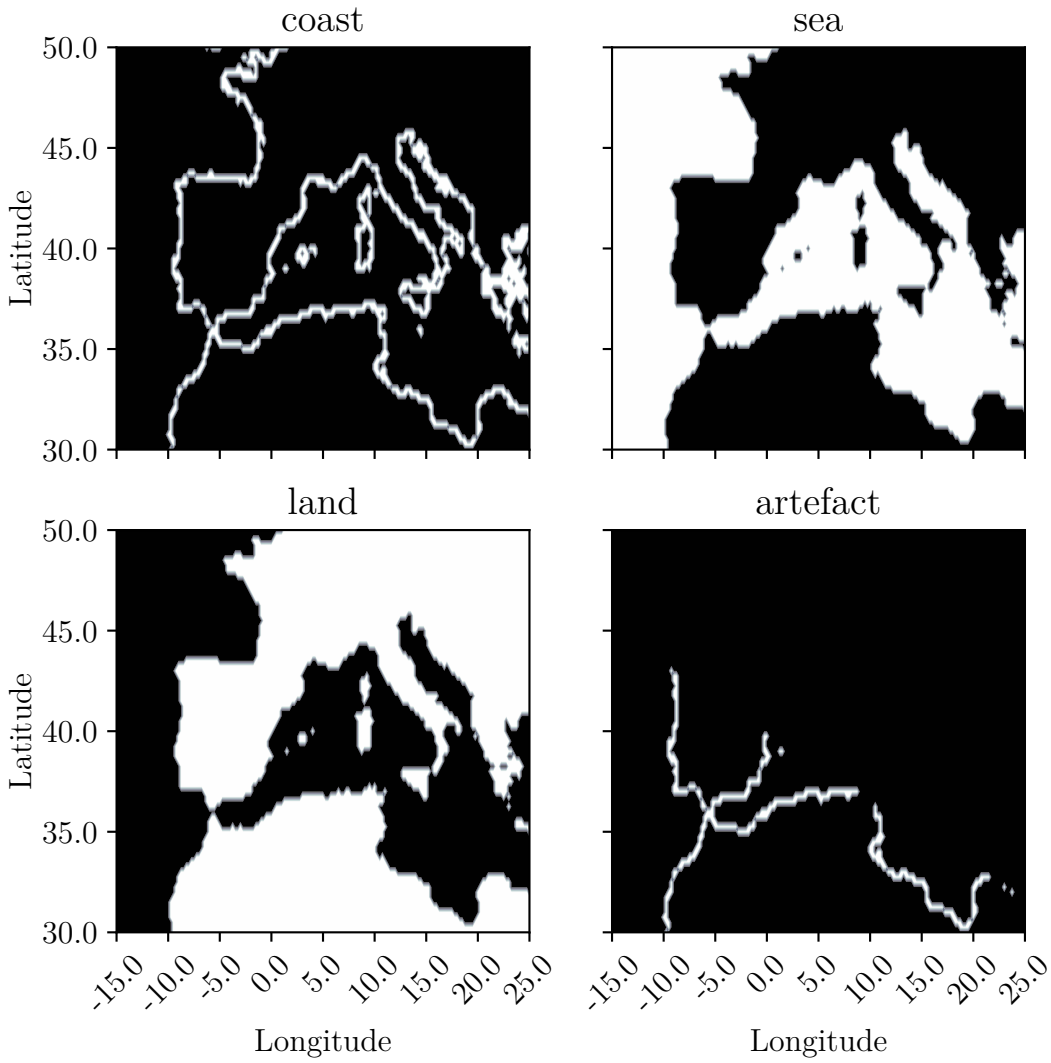


Figure 4.11: *Figure shows all filters, in white, available in the python package "sci-clouds".*

coastline is defined to be all pixels that are not either 100% land or sea. A threshold based binary classification is used to separate the coastline pixels into land or sea. The threshold is set to 50%. Pixels containing at least 50% of sea are most likely affected by maritime conditions, making it reasonable to classify them as sea pixels.

Based on a visual comparison to Figure 4.10, the artifact is defined to be all coastline pixels obeying the following inequality.

$$\theta + \frac{1}{3}\phi < 40 \tag{4.7}$$

All filters are displayed in Figure 4.11. No efforts have been made to remove the artifact

from the data.

4.1.7 Statistical Properties in ECC

To summarize and present the content of ECC, statistical properties are used. For clarity, the period used in this section is 2004 to 2018. The properties selected in this study are mean, minimum (min), maximum (max), median, Standard Deviation (STD) and Median Absolute Deviation (MAD). The most important results are presented in this section, and complementary figures can be found in the Appendix A. The data varies in space an time. Statistical properties are calculated over one or both of these axes.

Statistics over space and time

To study the statistical properties at different parts of the domain the filters, shown in Figure 4.11, were applied to the data before computing the statistical properties. Figure 4.12 shows a barplot summarizing the statistical properties for the five variables (Temperature, Surface Pressure, Cloud fractional cover, Specific and Relative humidity) and four filters (coast, land, sea and all). It shows minor differences between the statistical properties applied. The largest difference can be seen in the maximum value of relative humidity, where the maximum above land is larger than that above sea. The relative difference in magnitude between MAD and STD and the other variables from the reanalysis dataset, is solid. Most likely because the data is smoothed as a consequence of using assimilation model.

Temporal statistics

Figure 4.13 shows the spatially averaged monthly mean values for all variables. Seasonal effects and differences between land and sea are evident among all variables. For temperature and relative humidity, a more pronounced seasonal cycle over land, compared to sea, is evident. The remaining variables appear to have a small shift towards higher values over the sea. This is as expected, the pressure decrease with altitude and the sea is a source of humidity. On a monthly average the temperature over land is higher than the sea, but the difference is small, a couple degrees.

The spatially averaged time series for the first week of September in 2012 is shown in Figure 4.14, illustrating the relative strength of the signal fed into the input sequence. As a reference, the signals over land and sea are provided. All variables display diurnal variations. The figure shows small diurnal variations over sea, for all variables except clouds. In this week in September 2012, the temperature in ERA5 is remarkably constant. Varying within one degree and the temperature at sea, nothing at all. Remember that these variables stem from

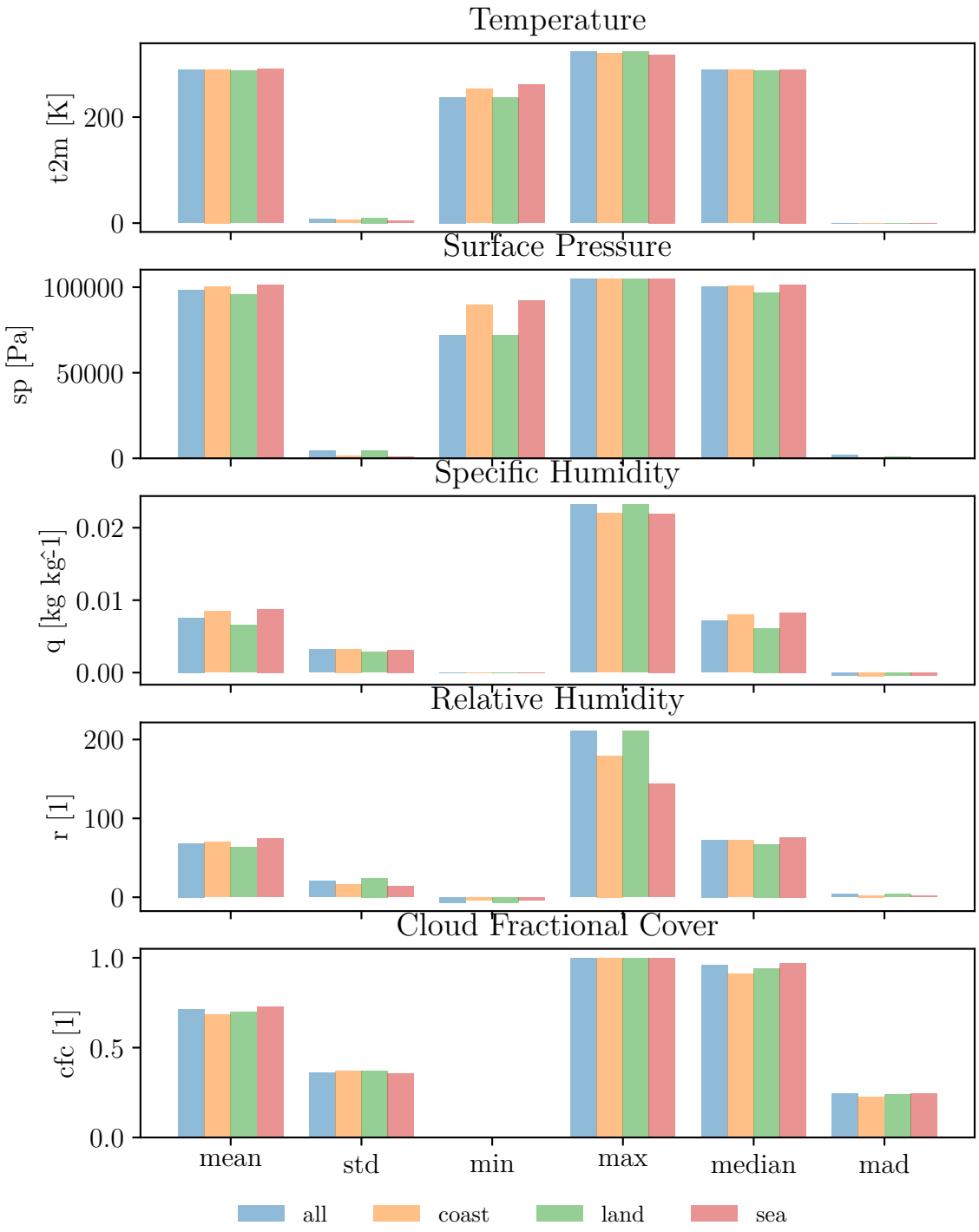


Figure 4.12: *Bar plot showing global statistics for different filters.*

the hourly reanalysis product, and the variables, except for clouds, show little variation, the amplitudes are nearly constant and the lines of values don't cross each other.

The complete series of first weeks of the remaining months of 2012 is shown in Appendix C. Under the assumption that the first week of a month is representative for that month. This

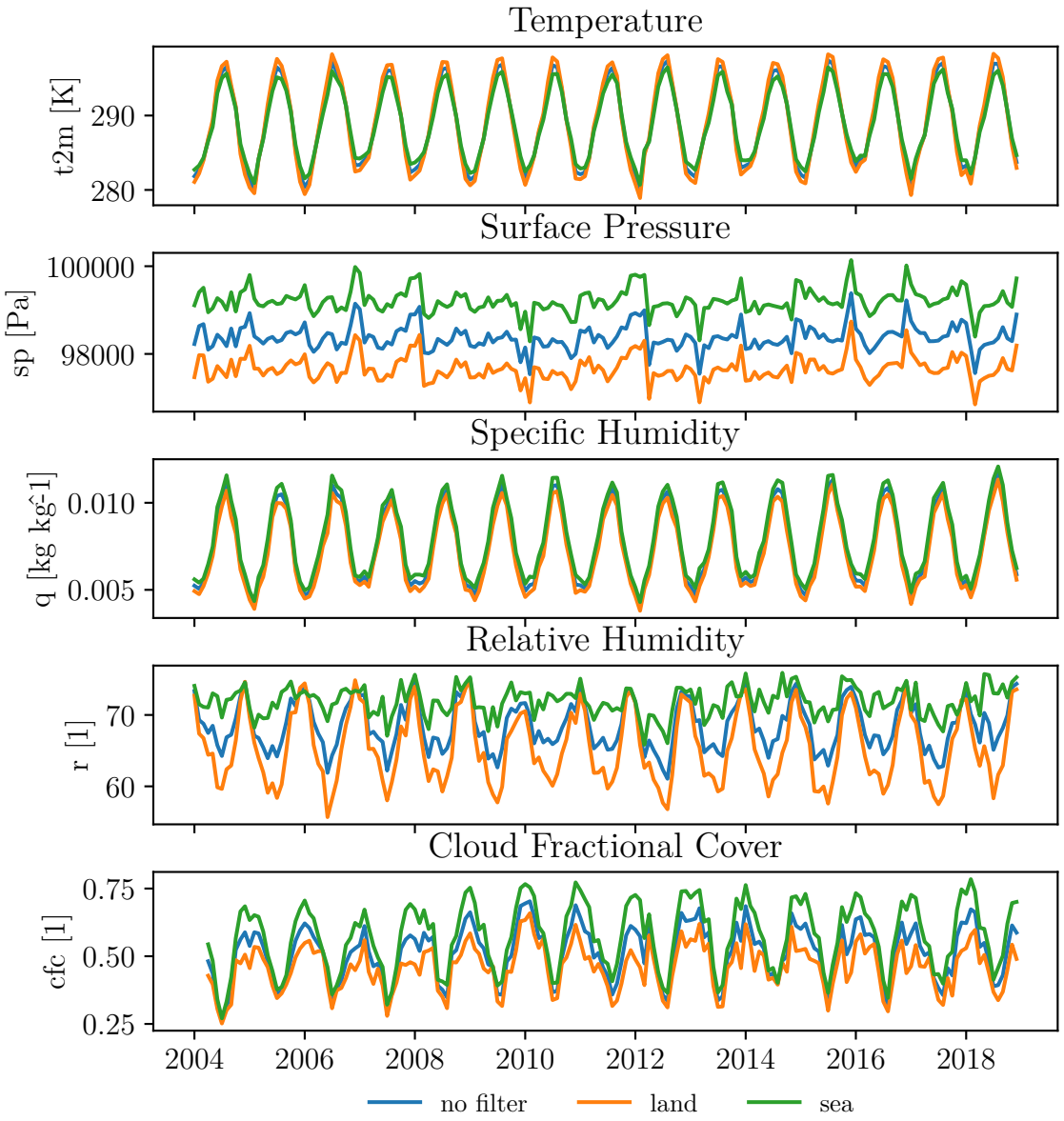


Figure 4.13: *Spatially averaged monthly values. Filters are applied for land and sea.*

series show that the diurnal changes of temperature and relative humidity vary most within the course of the year.

Note the spike appearing on September 2, 2012 in Figure 4.14. To examine this event Figure 4.15 was produced. From studying the evolution of the cloud cover it doesn't appear to be a numerical artifact. It is simply looks like there is a low pressure system centered in Europe, able to raise the cloud cover from 0.4 to 0.7 within one hour. Recall, that low pressure system swirls counter clockwise north of Equator, as mentioned in Section 4.1.2 .

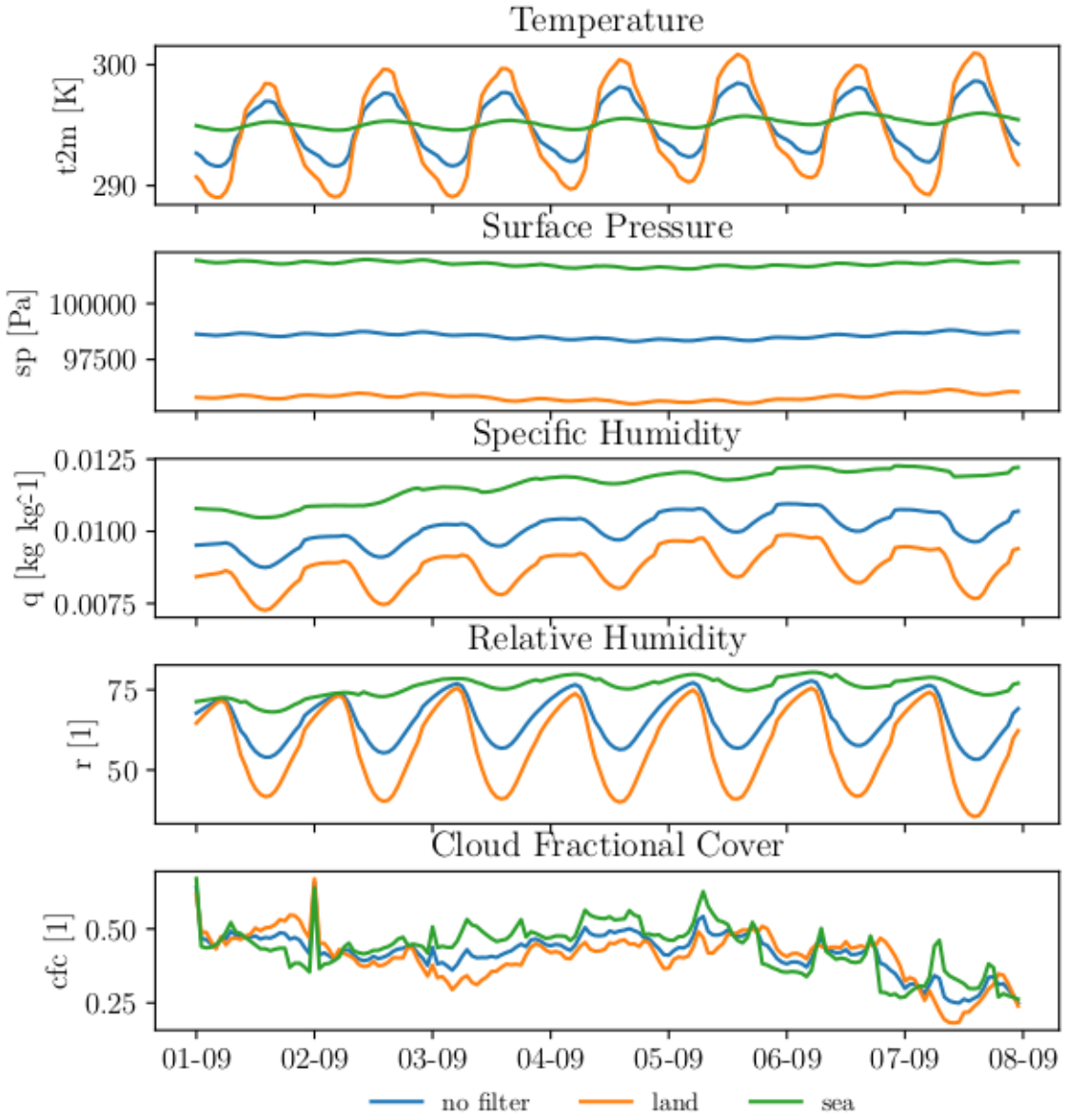


Figure 4.14: *Spatially averaged evolution of variables in ECC in the first week of September 2012.*

Temporal statistics

All the statistical properties computed over time for the variable CFC in ECC are plotted in Figures 4.16 and 4.17. Similar figures for the remaining variables in the dataset are presented in Section A.1. With the exception of minimum and maximum values, the cloud cover is affected by the coastline, on average there is a lower cloud fractional cover at the coast. In large regions the median value is one, revealing that in these areas more than half of the

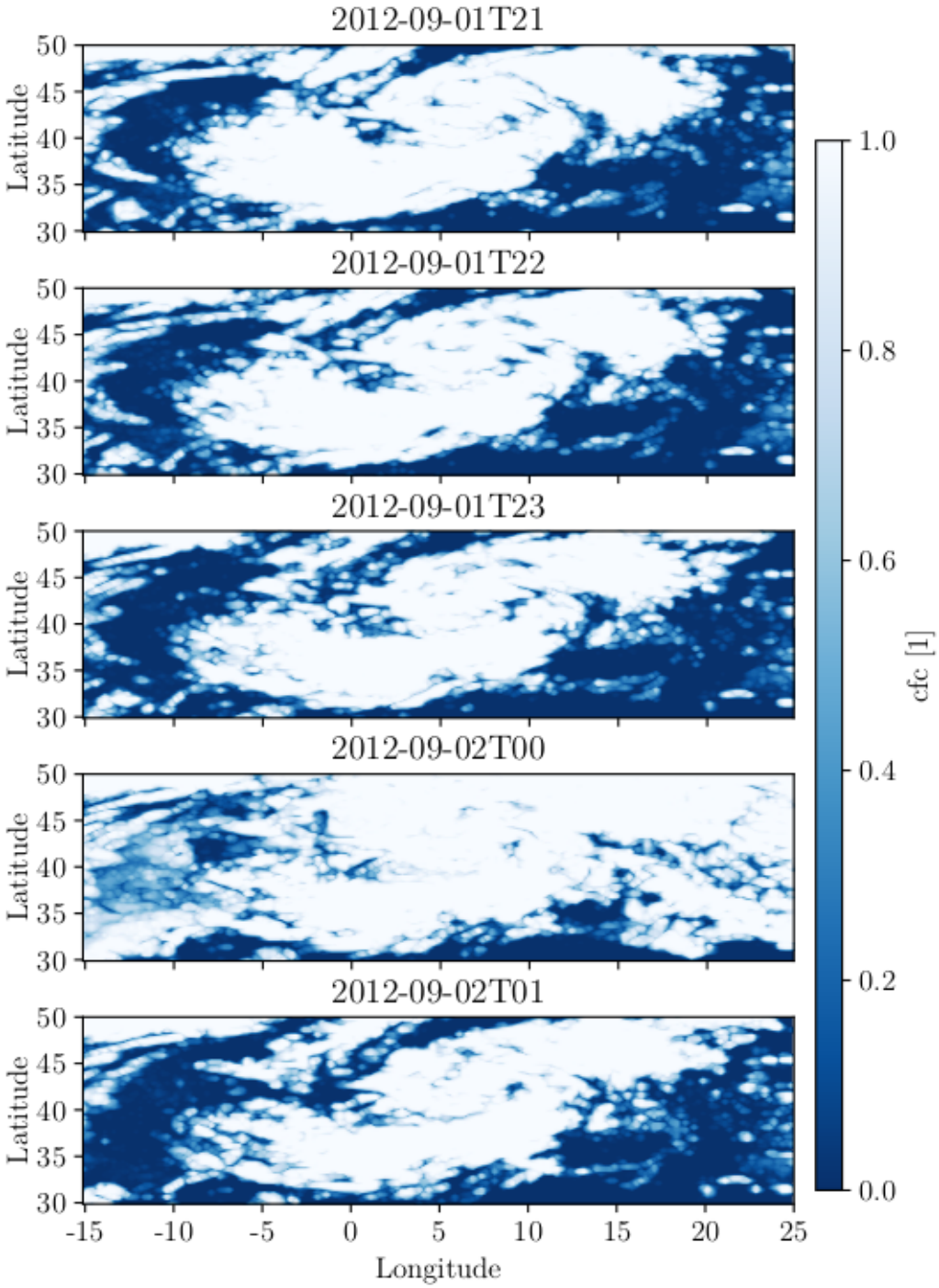


Figure 4.15: *Evolution of the spike detected in Figure 4.14.*

datapoints are 1.

The mean values displayed in Figure 4.16 show that on average the coastline has a lower cloud cover than the adjacent areas. This is supported by the bar plot, see Figure 4.12.

Figure 4.5 shows the patterns of magnitude of the areas contributing to a pixel. As expected,

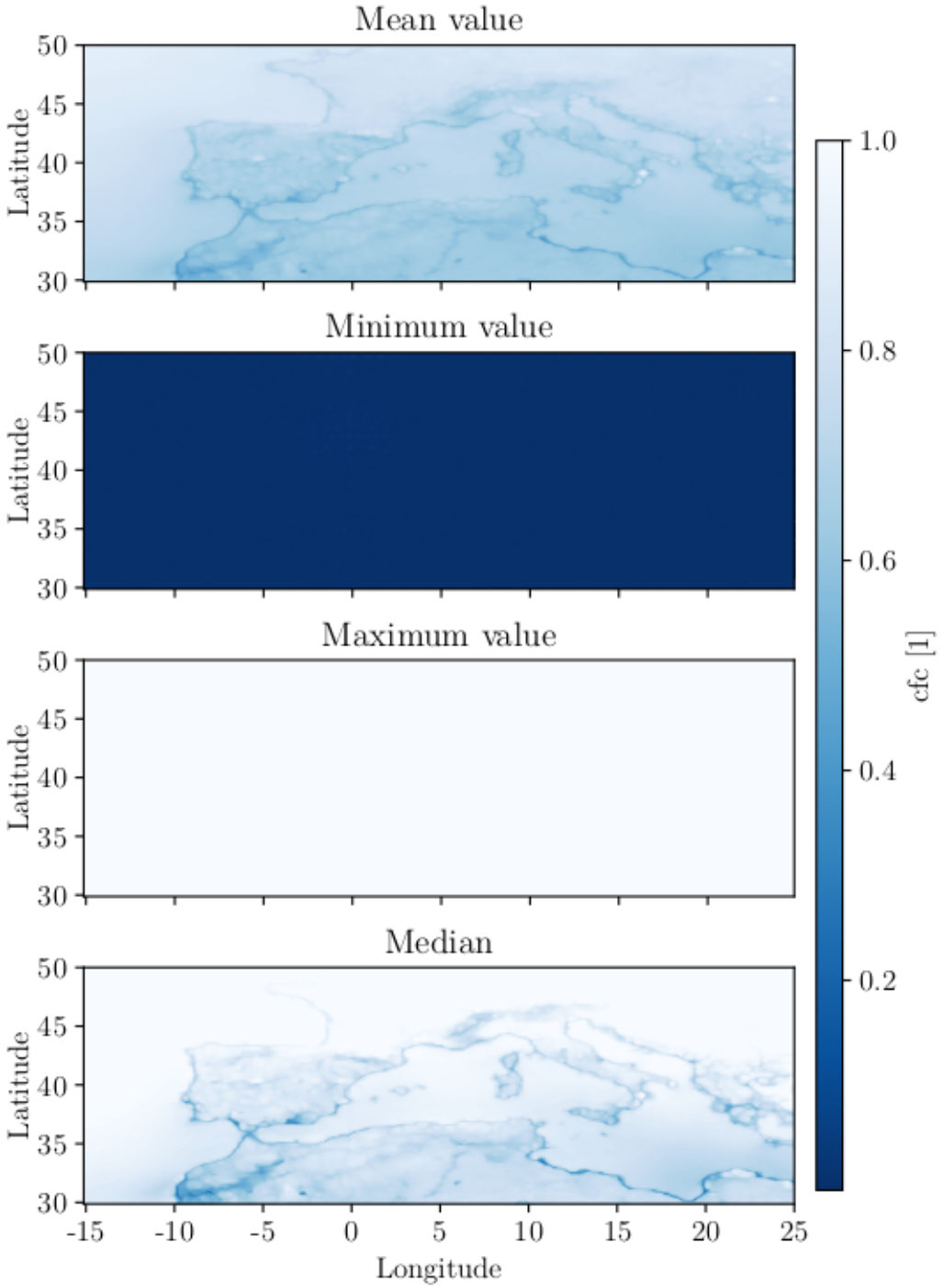


Figure 4.16: *Contour plot showing the local (pixel) statistics for cloud fractional cover.*

the minimum value is zero and the max is one. This tells us that in the course of a 14 year period, all pixel are cloudy and clear at least once. The standard deviation, shown in Figure 4.17, is higher over land. This indicate a larger variation in cloud cover in these regions. The median and MAD show similar patterns, this is not surprising since MAD is short for

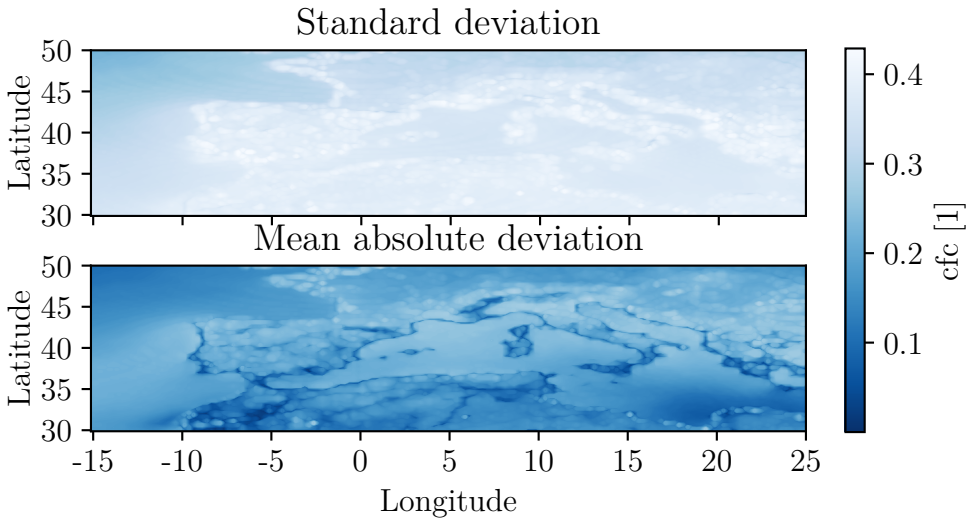


Figure 4.17: *Deviations in cloud fractional cover.*

Median Absolute Deviation, where the lowest values are found along the coast.

Correlation between cloud cover and environmental variables

Correlation describes how strongly a pair of variables are linearly related. A positive correlation tells you that an increase in one variable results in an increase in the other. A negative correlation describes the opposite connection, implying that an increase in one variable causes a decrease in the other.

The linear correlation coefficients from pairs of cloud cover and environmental variables such as temperature, pressure, relative and specific humidity are shown in Figure 4.18. Recall that all environmental variables are produced by a reanalysis. Pink illustrates negative correlation while green illustrates a positive correlation. Note that different patterns emerge from all variables.

Over land relative humidity is dominated by positive correlation with cloud cover, although some parts of Africa and the sea in the eastern Mediterranean have a negative correlation. The image of the surface pressure is remarkably similar, showing a similar pattern but with opposite signs. The negative sign seems reasonable, since high pressure is often associated with sinking motions in the atmosphere and clouds are formed by rising motions.

Specific humidity shows a clear shift at longitudinal degree 10. The land area in the west shows positive correlation with cloud cover while the eastern part shows a negative correlation.

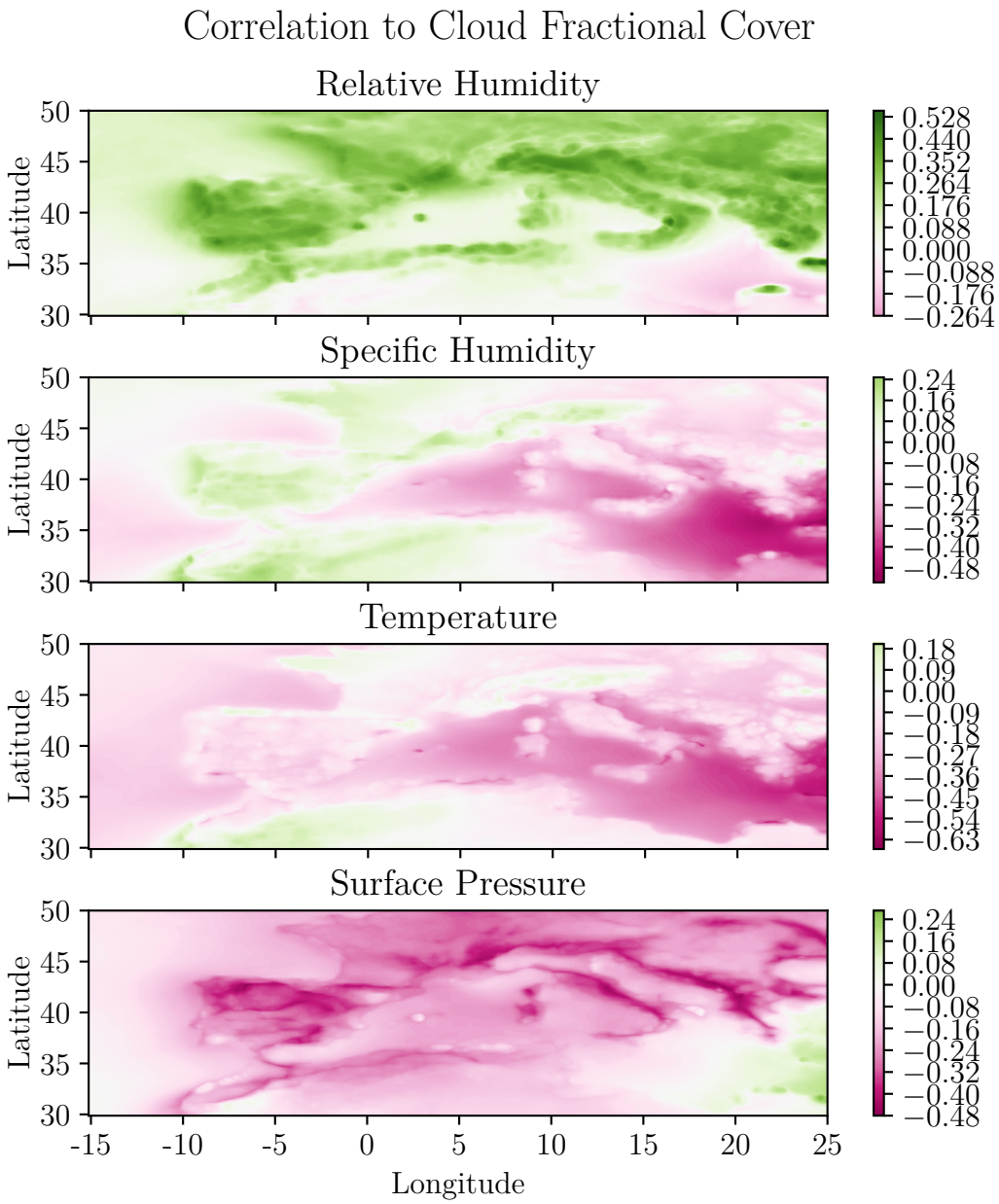


Figure 4.18: *Contour plot showing the correlation coefficient between environmental variables and cloud fractional cover.*

In most locations temperature is negatively correlated with cloud cover except in parts of the Alps, north coast of France and in north Africa. This seems reasonable since warmer air can retain more vapor.

4.1.8 Summary

ECC is comprised of five variables; temperature, pressure, cloud fractional cover, relative and specific humidity. These are collected from two sources; ERA5 and EUMETSAT. The

resolution available in ERA5 was preserved, while remapping the cloud mask to cloud fractional cover. The final product consist of hourly data on a 0.25^o uniform grid resolution in the period from April 2004 to December 2018. Cloud fractional cover (CFC) is produced from area weighting cloud masks. The AWRS is described in Section 4.1.3. The remaining variables are on their original format as provided by European Centre for Medium-Range Weather Forecasts (ECMWF). A summary of the original sources of the dataset is given in Table 4.1. More details on ERA5 is available in Section 2.5.1 and for the cloud mask in Section 2.5.4.

	ERA5		**MSG**	
	Type	**Variables**	**Type**	**Variables**
	Surface	2m Temperature Surface pressure	Satelite retrival	Cloud Mask
	1000 hPa	Relative Humidity Specific Humidity		
Projection	Uniform grid		Space-view grid	
Spatial resolution	0.25^o		-	
Output Frequencey	Hourly		15 min	
Availability	1979-onwards The periode 1950-onward is expected to be available for use sometime in 2020.		2004-onward	
License	Open Access. Need user from Copernicus Data Storage.		Researcher Licences to get 15min resolution. Open access (need user) at hourly resolution	

Table 4.1: *Summary of data used to compile ECC.*

4.2 Computer Experiments

This section describes the setup and configuration of the computational experiments conducted in this study, and the accompanied results. The first sections provide details on framework, structure and implementations. The following sections describes the experimental design for the ConvLSTM- and AR-models, focusing on the hyperparameters optimized. All models are evaluated on their ability to reproduce the cloud cover in the period from 2014 to 2018. At the end, the best configuration from each of the statistical models are evaluated on their ability to produce a 24-hour cloud cover forecast and compared against existing parameterizations in ERA5.

4.2.1 Framework, Structure and Implementation

The numerical methods used in this study are described in Chapter 3. The code is available on GitHub in the project repository named "MS" on https://github.com/hannasv/MS. Instructions for downloading reanalysis (ERA5) data using python is provided. MSG data is available via Earth Observation Portal on EUMETSAT's web pages. The dataset, ECC, is not published because of a licenses on the MSG data.

The code is developed in Python 3.7, a popular language for scientific software development. The source code is stored in the package *sciclouds*, made available on GitHub through the project repository. Developed modules draw inspiration from the structure of *scikit-learn* (Buitinck et al., 2013). The ConvLSTM is implemented using Tensorflow's keras API (Martin Abadi et al., 2015) which simplifies many aspects of building and executing machine learning models. To utilize the analytical solution the AR-models are trained and evaluated using self-implemented modules. The python package "sciclouds" provides a self-implemented version of AR-models, using the analytical solution to the least squares problem derived in Section 3.3.

Visualizations are generated using *Matplotlib* (Hunter, 2007), *Seaborn* (Waskom et al., 2014) and maps using the package *Cartopy* (Met Office, 2015). Other illustrations are developed using TIkZ, a language used for producing technical illustrations within the environment of LaTeX.

The package versions are documented in the *requirements.txt* and the project environment called "sciclouds" is ready for installation. This is a conda environment, the yaml-file lists the Python packages and requirements necessary for running this code. Below you find the

code example for cloning the project and installing the environment.

```
git clone https://github.com/hannasv/MS.git
cd MS
conda env create -f environment.yml
conda activate sciclouds
python setup.py install # installing package from source
```

Supplementary material for remapping satellite data and filtering masks is available in the supplementary repository https://github.com/hannasv/MS-suppl. The filters are generated from within the environment of PyAEROCOM (MET Norway, 2020).

4.2.2 Hardware

The experiments described below, are conducted on a DGX-2 system consisting of 16 NVIDIA Tesla V100 GPUs, each of 32Gb local memory and 1.5Tb shared memory. The data is stored on a Remote Direct Memory Access (RDMA) accessed over Infiniband. The DGX-2 system is designed for a high level concurrency and scheduling workers competing for system resources. More detailed hardware specifications is provided in Table 4.2.

Device	Type
GPU	Tesla V100-SXM3-32GB
CPU	DualProcessor AMD Epyc7601 (SMT2) w/2TB ram and 4TB NVMe

Table 4.2: *Hardware specifications for the test environment used on eX^3. The operating system is Ubuntu 18.04.4.*

4.2.3 Training, validation and test split

Gradient methods are at the heart of every machine learning algorithms. This type of optimization is based on the principle that the model is continuously evaluated against the validation dataset and weights are adjusted to reduce the loss. This raises the need for two datasets during training. The AR-models are computed based on an analytical solution and have no need for the extra data set.

Based on the assumption the most recent partition is most representative for the near future, both models were tested on the period 2014 including 2018. The AR-model is trained on the

period 2004 to 2013, while the ConvLSTM-model is trained on 2004 to 2011 and validated on 2012 to 2013.

Other notable differences in the input data is related to handling missing values. The AR-models use shorter sequences, and samples containing missing values are simply removed. The ConvLSTM-model utilize longer sequences, and missing values are replaced by the out-of-sample value, $c = 1.5$.

4.2.4 Autoregressive models

Traditionally an AR-model describes time-varying processes as a linear combination of its previous values, and for some models in this study it also includes other meteorological values. An AR-model is composed of 13041 individual regression models, one for each grid box (see Section 3.3).

In the search for the best model configuration, four hyperparameters have been tuned. These are feature scaling of the predictors, the inclusion of bias, number of lags and a potential inclusion of environmental variables. Varying combinations of these parameters results in the set of models trained in this study. The optimization strategy used is starting with the simplest models and gradually increasing the complexity.

Feature Scaling

Feature scaling is used to standardize the predictor variables.

$$\mathbf{x} = \frac{\mathbf{x} - \bar{\mathbf{x}}}{\mathrm{STD}(\mathbf{x})} \tag{4.8}$$

The transformation is computed by applying Equation (4.8) to the predictors, represented by $\mathbf{x}$, $\bar{\mathbf{x}}$ represent its average and STD its standard deviation. The resulting data has a reshaped distribution resembling a standard normal distribution, with zero mean and unit variance. This offers an additional benefit of increased numerical stability.

It is important to perform the transformation after the data is split into training and test. The mean and standard deviation should be computed based on the training set and applied to both sets. The model is trained to find relations in transformed data. Consequently the test data need to be transformed before the model can be evaluated.

The partitioning of datasets prior to the transformation is necessary to avoid a information leak between the test data and the trained model. If it was done differently it would result in an unrepresentative measure on performance.

Lags and Environmental variables

The dataset prepared for a particular model is determined by the number of lags and the inclusions of environmental variables (temperature, surface pressure, relative and specific humidity). This controls a models the degrees of freedom. All models are trained either on the full set of environmental variables or none of them. They never appear in isolation.

Lags describe the number of previous timesteps of CFC included as a predictors. For example, if the lag is three, $L = 3$, then the CFC is predicted based on the cloud cover for the previous three hours. Utilizing all time steps up to and including three hours back in time.

Experimental setup

The naming conventions for the AR-models used in this study AR_{TB_L} or TR_{TB_L}. AR or TR describe whether the environmental variables are included in the dataset of not. TR is short for traditional and represents the case when environmental variables are omitted. AR represents the opposite, their inclusion. B represents bias, T symbolizes that scaling predictors is applied and L_x reveals the number of lags, represented with x.

To ease the understanding of the naming convention used, Table 4.3 provide four examples. Applying the following convention, $\times$ denoted not applied, ✓denotes applied. The hyperparameters bias and transformation of the predictors is mutually exclusive. Applying both have no benefits as the transformation cancels the effect of the bias when subtracting the mean.

	Feature Scaling	**Lag**	**Environmental Variables**	**Bias**
TR_{B_1}	$\times$	1	$\times$	✓
AR_{B_0}	$\times$	0	✓	✓
AR_{T_1}	✓	1	$\times$	$\times$
AR_{B_4}	$\times$	4	✓	✓

Table 4.3: *Example configuration of AR-models, where $\times$ denoted not applied, ✓denotes applied.*

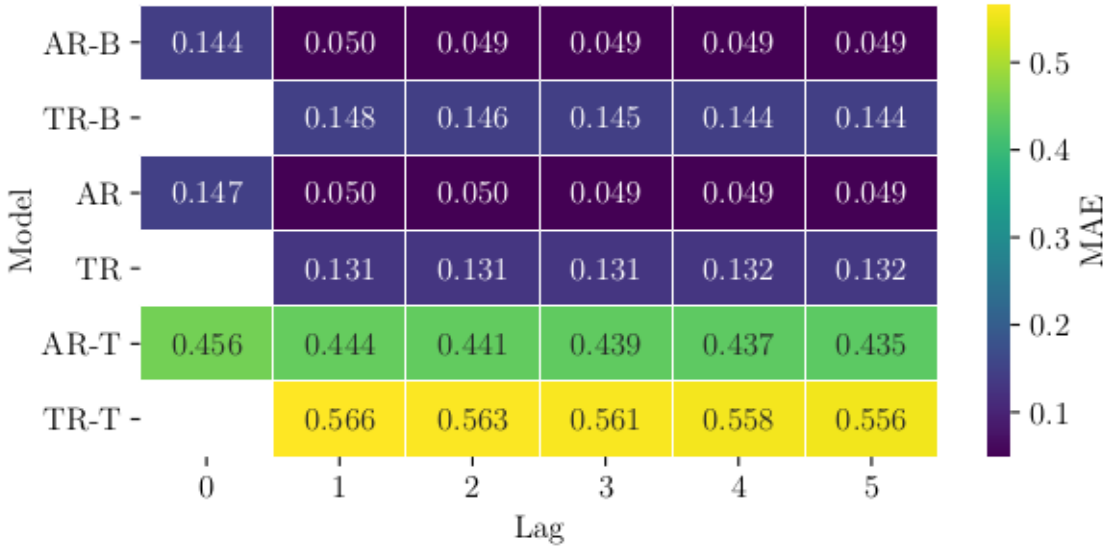


Figure 4.19: *Heatmap showing the area averaged test MAE for all the AR-models included in this work. AR represent the inclusion of environmental variables, TR represent the opposite, B is bias and T is the feature scaling, transforming the distribution of the input data.*

Evaluation

The models are all evaluated using Mean Absolute Error (MAE) and the AR-models are optimized to fit the next timestep, not longer sequences. Figure 4.19 shows the performance of all AR-models included in this study, varying the number of lags on the first axis and the other hyperparameters on the second axis. The score is computed by using Equation (3.19), and putting the parameters $m = 81$, $n = 161$ and $k = 43824$. With the exception of the $AR - T$-configuration, most models increase most rapidly in performance when adding the first lag. The performance continues to increase for larger numbers of lags, but at a much slower rate. This indicates that the cloud cover at previous timesteps is indeed a useful predictor. The largest variations in performance is caused by varying configurations of AR, B, T and TR.

The configurations employing feature scaling, T have the overall lowest performance. A grid MAE of roughly 0.5 is high when the target varies in the range from 0 to 1. The inclusion of a bias in combination with AR improves the performance, while for TR it has the opposite effect and decreases the performance. In conclusion, T is not a setting suitable for the cloud forecasting problem.

The TR-configuration performs a lot better than T, and the set of models have an MAE close to 0.14. Since AR outperforms TR for all configurations, except for T, this indicates that the environmental variables provide useful information.

Relatively small gain, may still be important, and the skill of the best model, $AR - B - L_5$, is 0.04901, which is excellent. This shows that there is enough information in the set of environmental variables and previous cloud cover to predict cloud cover one hour into the future.

Parameterizatin cloud cover using $\mathbf{AR - B - L_5}$

Figure 4.20 shows the weights in $AR - B - L5$. Note that the colorbars are different for all subplots, but the colors are consistent, with red being positive and blue negative. The AR-model is a weighted sum over all the variables. Negative input values are unphysical. With the exception of r, negative values are not present, as documented in Section A.1. In the case of relative humidity, r they are rarely present, but exist, and the minimum values is -6.6505. The surface pressure weight is negative for the entire grid. Higher values of surface pressure will cause large reductions in cloud cover. This is in agreement with the cloud physics described in Section 4.1.2, stating that areas of high pressure are associated with descending airmasses, which do not lead to cloud formation.

For the other variables, positive values contribute to cloud formation and negative values to dissipation. By studying the weights of the five hours of previous timesteps, it is clear that they all contribute in producing the predicted cloud cover. L_1 have the highest weights, reaching a maximum of 0.9, and is nearly constant over the entire grid. In comparison the weights of other lags are small, and this may explain the minor change increasing number of lags, shown in Figure 4.19.

The behavior of relative and specific humidity is puzzling. They exhibit the approximately opposite behavior. North Africa has one sign, and Europe has another. In most regions where relative humidity appears to promote clouds, specific humidity appears to reduce the cloud cover. The specific humidity has values of $\hat{q}$ 0.005 see Figure A.4 and for relative humidity values are $\hat{r}$ 75, which makes them the same order and the weighted sum can potentially cancel.

In most areas of the European continent temperature is a positive indicator for cloud formation. The temperature weight also shows an opposite spatial distributions compared to that

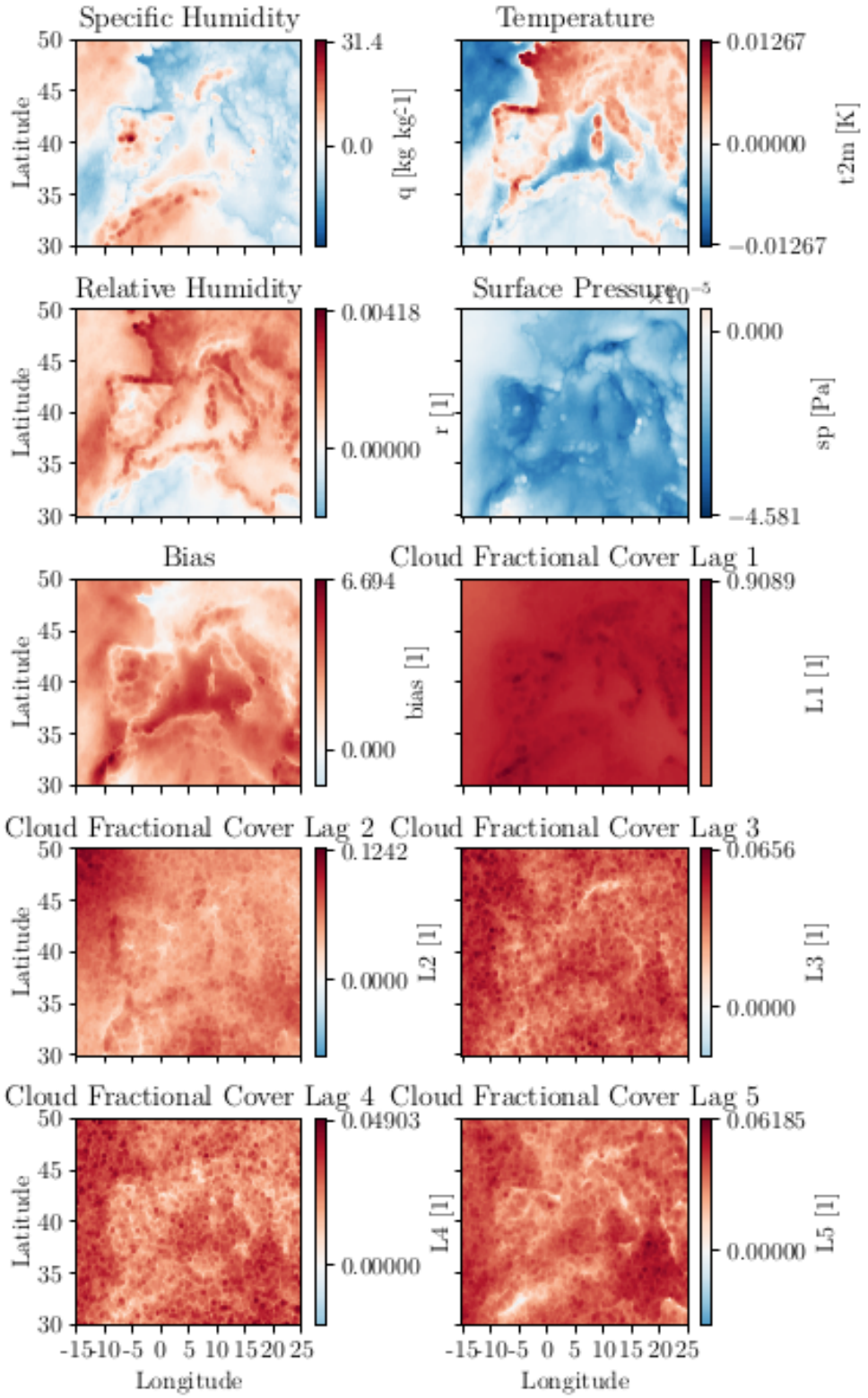


Figure 4.20: *The weights of the $AR-B-L_5$-model.*

of specific humidity, thereby resembling the weight of relative humidity.

4.2.5 Convolutional LSTM

The formulation of the air quality forecasting problem presented by Sun and Liu, see Section 3.4 is similar to the formulation of the cloud fractional cover forecasting problem presented in this study. This study adopts the machine learning setup in Sun and Liu (2018) . Manual tuning of the models is applied to avoid a breakdown of the computer caused by too many parameters. When building ConvLSTM networks, the list of tunable parameters is extensive. These hyperparameters are divided into subsets of tuned and constant parameters.

The following section describes the tuned hyperparameters, batch size, sequence length, number of hidden states and the dimensions of the filter. The dataset is partitioned into subsets called batches. The batch size is the number of sequences a weight update is based on. Epochs describes the number of times the model loops over the entire dataset. The sequence length is the number of timesteps a model is optimized to learn to predict. The number of hidden states it the number of kernels it learns in each layer. The kernel dimensions determines the number of neighbors influencing an activation. Using 1x1-filter results in the state-to-state transitions similar to AR-models by removing interactions between adjacent pixels. A more detailed description on these parameters is provided in Sections 3.2.1 to 3.2.4.

This section describe the hyperparameters kept constant. A model consist of a set of ConvLSTM-layer, between each layer there is a Batch Normalization (Batch Norm)-layer (implemented using default settings). This type of layer was introduced by Ioffe and Szegedy (2015) first implemented for a CNN. The results showed three benefits, the network was less sensitive to the weight initialization, higher learning rate and it did not need dropout. *Dropout* is another hyperparameter, which randomly removes some of the trained weights to prevent overfitting. This is computationally expensive, disabling dropout accelerates the training process. “Padding same” is applied to all ConvLSTM-layer, to make sure the input and output dimensions are the same, see Section 3.2.5. The model returns a sequence and the input sequences are not shuffled. The dimensions of the output layer are determined by the task at hand. To produce a cloud cover forecast the output kernel and number of hidden states need to be one.

The weights were initialized based on the scheme “LeCun uniform” (Lecun et al., 1998).

Callbacks such as terminate on NaN's have been applied to avoid prolonged training time. The optimizer ADAM is used with the following settings, learning rate $= 0.001$, beta1 $= 0.9$, beta2 $= 0.999$, epsilon $= 1e-07$ (Kingma and Ba, 2015). The default settings in Tensorflow use epsilon $= 1e-08$. The loss function is Mean Squared Error (MSE), and the models are evaluated based on Mean Absolute Error (MAE). Both metrics are described in Section **??**. A list of compiled non-trainable architecture are included in the Appendix D as a reference point for further studies.

Experimental setup

Models are given names based on an extension of the convention from Shi et al. (2015) . The batch size and sequence length is included and the resulting naming convention is $ConvLSTM - B_x - SL_y -$ hidden states $- filter \times filter$. Table 4.4 provide a set of example configurations, here the square brackets list the number of hidden states (or kernels), and the position of the layer. If the bracket has three members the network has three layers.

ConvLSTM Model	Sequence Length	Batch Size	Hidden States	Kernels
$B_{10} - SL_{24} - 16 - 3 \times 3 - 16 - 3 \times 3$	24	10	[16, 16]	[3, 3]
$B_{10} - SL_{24} - 32 - 1 \times 1 - 32 - 1 \times 1$	24	10	[32, 32]	[1, 1]
$B_{10} - SL_{24} - 32 - 3 \times 3 - 32 - 3 \times 3$	24	10	[32, 32]	[3, 3]
$B_{10} - SL_{24} - 32 - 5 \times 5 - 32 - 5 \times 5$	24	10	[32, 32]	[5, 5]
$B_{10} - SL_{24} - 8 - 3 \times 3 - 8 - 3 \times 3 - 8 - 3 \times 3$	24	10	[8, 8, 8]	[3, 3, 3]
$B_5 - SL_6 - 32 - 3 \times 3 - 32 - 3 \times 3 - 32 - 3 \times 3$	6	5	[32, 32, 32]	[3, 3, 3]

Table 4.4: *Examples of -model names and their configurations.*

Evaluation

The input volume of ConvLSTM-models are different from the AR-model, and the axis "batch" and "sequence length" are merged before the score is computed by using Equation (3.19), and putting the parameters $m = 81$, $n = 161$ and $k = 43680$. Note that the k value is a bit smaller than for AR-models, this is caused by employing "drop remainder batch" during training. Figure 4.21 shows the loss curves in the training process for all the compiled models in this study. As expected, they all "learn" the most rapidly in the beginning of the training process. This is shown in the figure as the steep drop in loss between the first and second epoch.

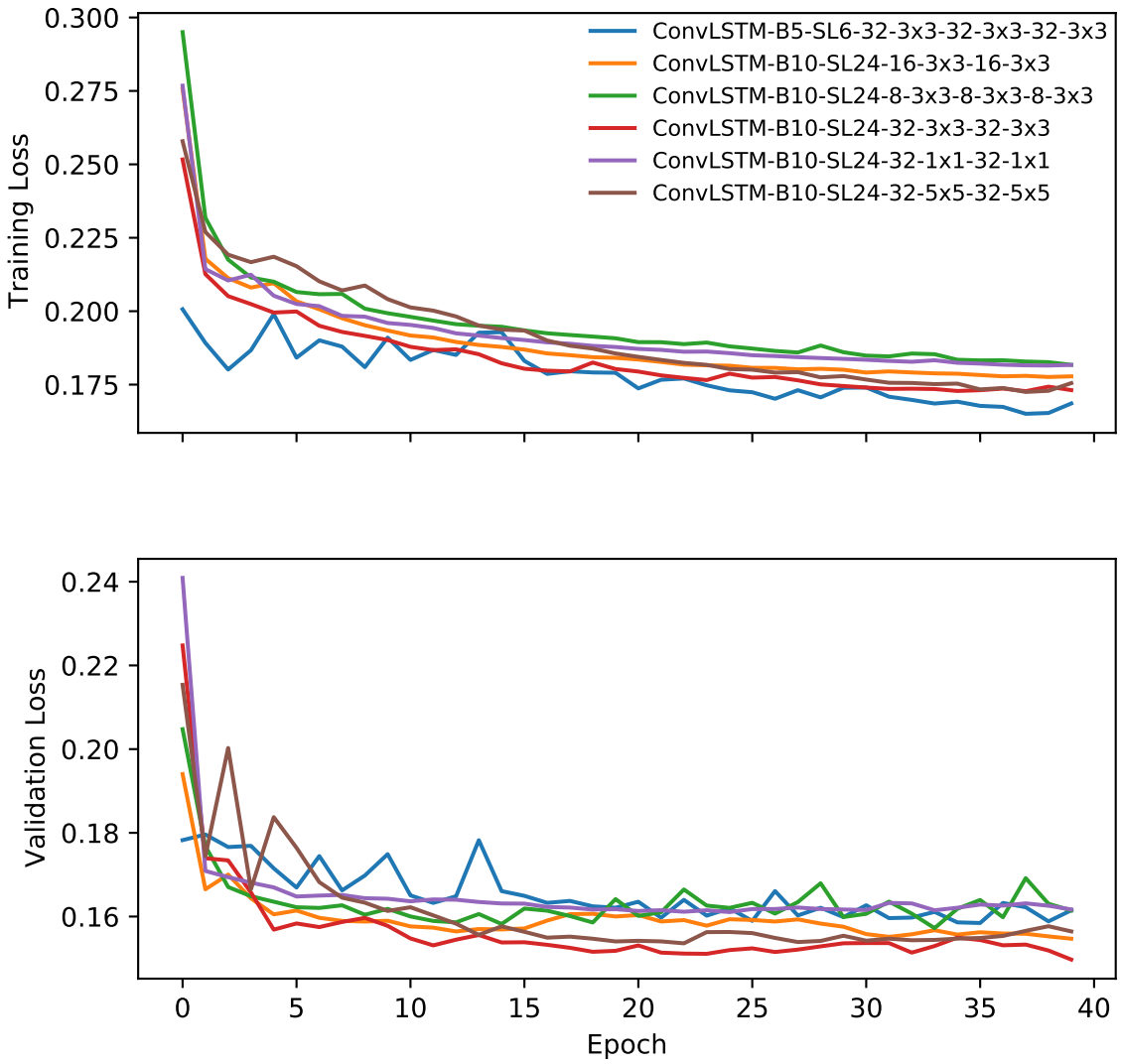


Figure 4.21: *The loss of the trained model as a function of epochs.*

From Figure 4.21 and Table 4.5 it is clear that the best performing ConvLSTM-model is the $ConvLSTM - B_{10} - SL_{24} - 32 - 3 \times 3 - 32 - 3 \times 3$. The run can be seen as the red line in both the figure and table. It has the lowest test and validation loss after 40 epochs, though another model $ConvLSTM - B_5 - SL_6 - 32 - 3 \times 3 - 32 - 3 \times 3 - 32 - 3 \times 3$ has a better training loss. As might be deduced based on the similarities in the names, the two models have similar architectures. The last model has an additional layer, this may have allowed it to learn a better representation on the training data presented, however the skill of predicting on unseen data is most important. The best configuration is therefore $ConvLSTM - B_{10} - SL_{24} - 32 - 3 \times 3 - 32 - 3 \times 3$.

ConvLSTM Model	Train Loss	Val Loss	Test Loss	Num. Params.
$B_{10} - SL_{24} - 16 - 3 \times 3 - 16 - 3 \times 3$	0.1779	0.1547	0.1575	30 296
$B_{10} - SL_{24} - 32 - 1 \times 1 - 32 - 1 \times 1$	0.1817	0.1617	0.1649	13 464
$B_{10} - SL_{24} - 32 - 3 \times 3 - 32 - 3 \times 3$	0.1731	0.1497	0.1534	115 864
$B_{10} - SL_{24} - 32 - 5 \times 5 - 32 - 5 \times 5$	0.1755	0.1564	0.1589	320 664
$B_{10} - SL_{24} - 8 - 3 \times 3 - 8 - 3 \times 3 - 8 - 3 \times 3$	0.1817	0.1615	0.1634	12 920
$B_5 - SL_6 - 32 - 3 \times 3 - 32 - 3 \times 3 - 32 - 3 \times 3$	0.1686	0.1615	0.1633	189 848

Table 4.5: *Results, metrics and number of parameters for the trained models. The best model ConvLSTM-model is highlighted in light blue. The loss presented is averaged over on batch, this is the keras default. Since its only used to chose the best model, there is no reason to upscale the numbers.*

In some cases reducing the spatiotemporal resolution can enable the model to learn even more (Shi et al., 2015). This is arguably not applicable for this task, since cloud cover has an average lifetime of one hour, as mentioned in Section 2.1. The reduction can not be applied without most likely producing a significant loss of information.

Comparisons of input data to the works by Shi et al. (2015) and Sun and Liu (2018), has to be done based on input volumes. The dimensions are flattened to generalize the comparison. Shi et al. (2015) is trained on $1,629,600,000$ data points, Sun and Liu (2018) on $28,513,800$ and ECC on $21,220,315,200$. In other words, this study has trained a ConvLSTM-model on a much larger amount of data than earlier studies.

$ConvLSTM - B_{10} - SL_{24} - 32 - 3 \times 3 - 32 - 3 \times 3$-model architecture is shown in Figure 4.22. The model consists of three Batch Norm (gray) and ConvLSTM (cyan) layer pairs. Both ConvLSTM layers have 32 hidden states and a 3×3-filter (blue). The output (cyan) layer has one hidden state and the filter (blue) dimension of 1×1. The input (green) shape is $10 \times 24 \times 81 \times 161 \times 4$ and output (red) $10 \times 24 \times 81 \times 161 \times 1$.

Figure 4.23 illustrates the finer structures within the input batch. It is impossible to show the entire five dimensional input volume in one sketch, so for illustrative purposes the two first sequences, located in the first batch, is shown. The first dimension is the batch (green),

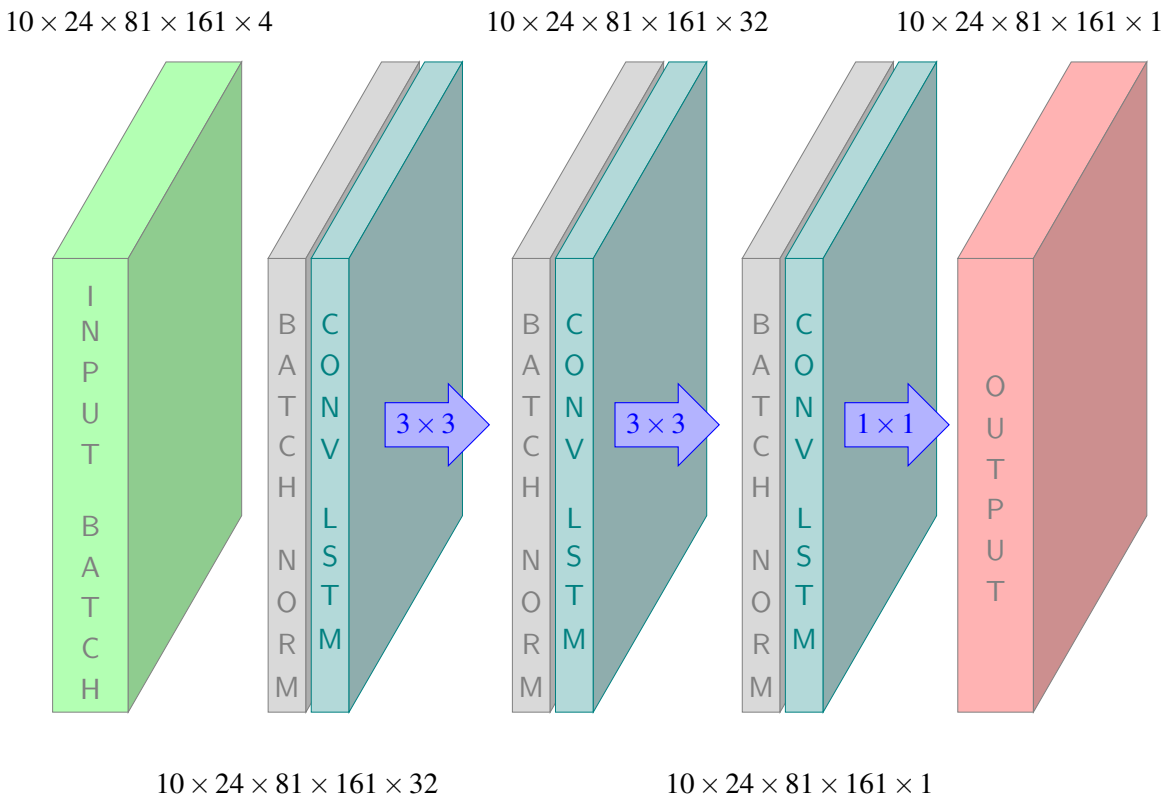


Figure 4.22: *The architecture of the cloud cover forecasting model developed in this study.*

the second is the sequence length (pink), the third is latitude, fourth is longitude and fifth is the number of environmental variables (illustrated in layers of different colors). A sequence consist of 24 weather data volumes, here indicated by the timestamp. A weather data volume has the dimensions $81 \times 161 \times 4$, the last dimension the environmental variables, the other dimension are the latitude and longitude.

4.2.6 Temporal Performance

This section contains the evaluation of the $ConvLSTM - B_{10} - SL_{24} - 32 - 3 \times 3 - 32 - 3 \times 3$, $AR - B - L_5$ and ERA5 against ECC. The metric used is MAE and the period is from 2014 to 2018. In making this comparison, please keep in mind that the AR-model is optimized to fit and evaluated on its ability to predict one timestep. The ConvLSTM-model is trained and evaluated on its ability to fit a sequence of 24-hours. Some of the dependent variables in ERA5 is assimilated against radiences from MSG (Hans Hersbach et al., 2020).

Figures 4.24, 4.25 and 4.26 display the temporal skill of the parameterization. They all show different regional biases and clear distinction between land and ocean, as the outline

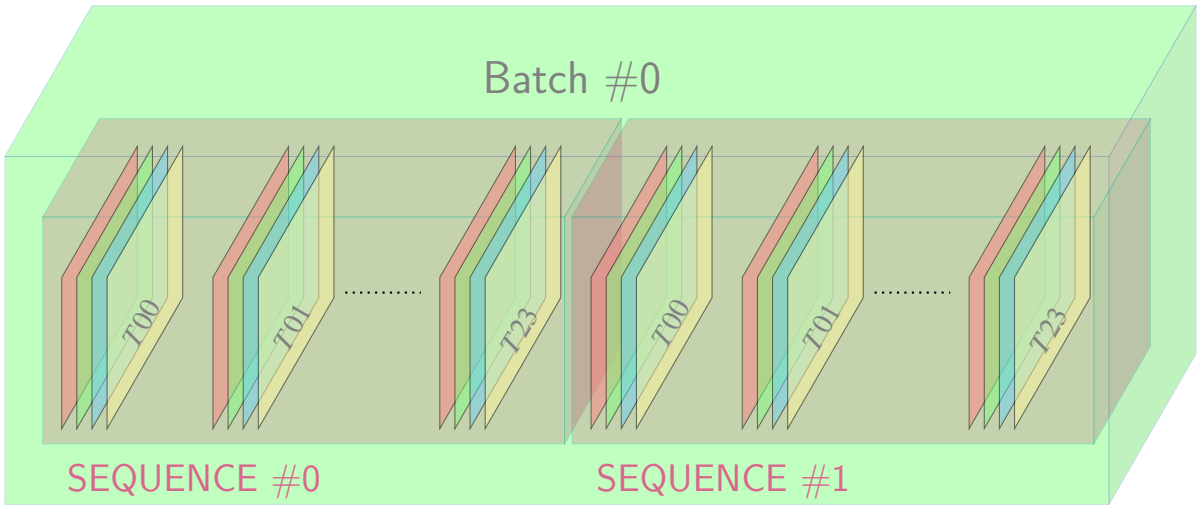


Figure 4.23: *A subsection of the input volume to the* $ConvLSTM - B_{10} - SL_{24} - 32 - 3 \times 3 - 32 - 3 \times 3$*-model. Illustrating the content of the two first sequences in the first batch. The input volume is divided into batches (green), each batch is divided into sequences (pink) and each sequence contains 24 weather datavolumes, illustrated using four layers of different colors, one for each input variable. The weather data volumes are labeled with the timestep,* T*.*

of Europe and North Africa is clearly visible. At first sight, the ERA5 has a more spotted pattern than the others. All models get a high skill in the Atlantic off the coast of Spain and France. The same pattern can be found in Figure A.6 showing the standard deviation in surface pressure, but it is most likely not related.

The regions of low biases in Figures 4.26 and 4.25 are the same regions exhibiting a low mean and median cloud cover as shown in Figure 4.16.

Both these representations of CFC have trouble with the Nile Delta. This does not seem to be the case for $AR - B - L_5$. The correlation between ECC and relative humidity is high compared to adjacent regions, as shown in Figure 4.18. It is also worth mentioning that the bias and weight for r is high in this region, which might explain the superior performance of AR.

Unfortunately it was only recognized towards the end of this study that some models had issues when training. These problems were partly numerical issues related to matrix inversions and partly related to corrupt files. It is reason to believe this occurred during training since other configurations have results for all pixels. The temporal performance of other

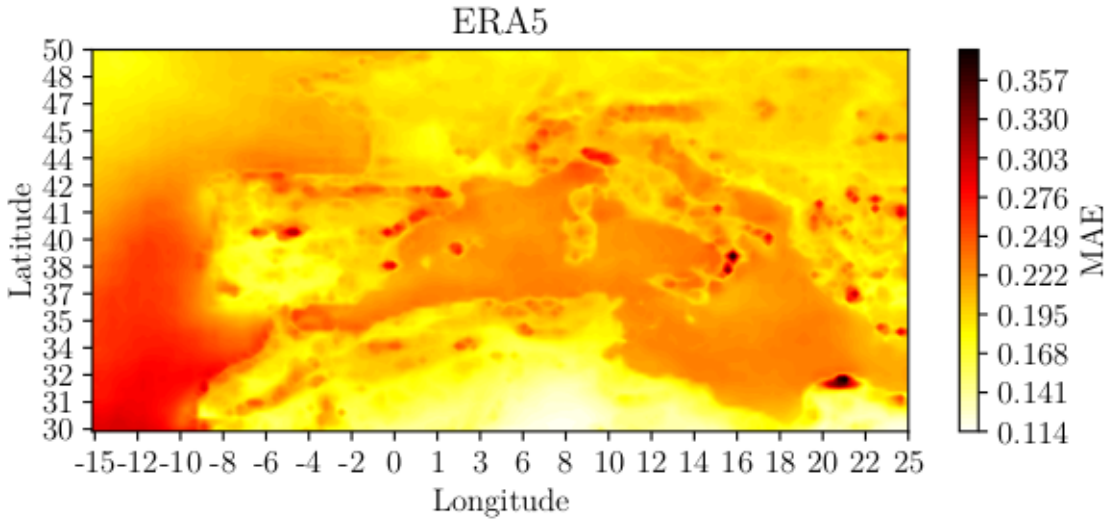


Figure 4.24: *Evaluation of the parametrization in ERA5.*

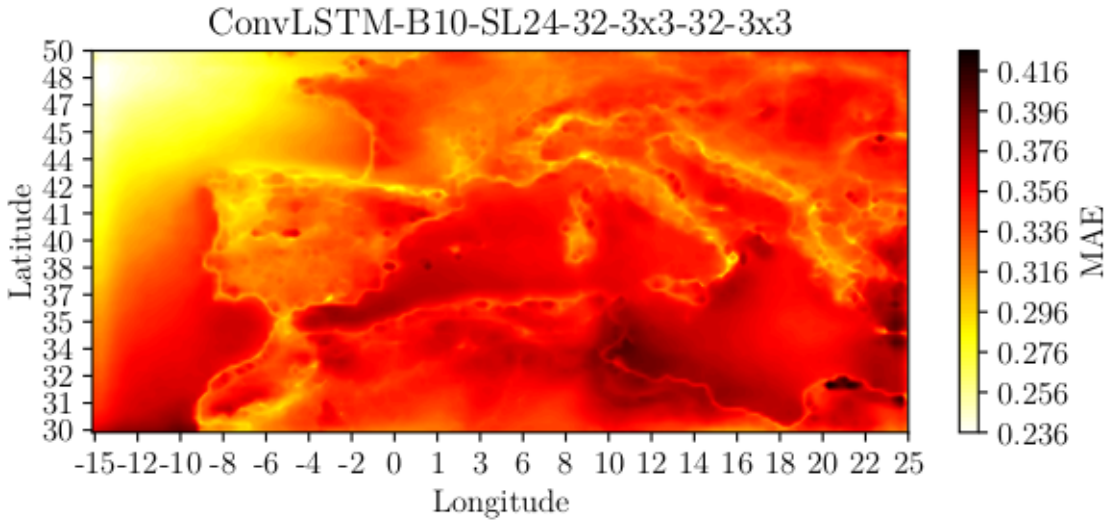


Figure 4.25: *Evaluation of the parameterization made by the* $ConvLSTM-B_{10}-SL_{24}-32-3\times3-32-3\times3$*-model.*

AR-models can be found in the Appendix D.1.

Table 4.6 summarizes the spatially averaged performance of the models. These results indicate that the overall best parameterization is made by the $AR-B-L5$-model.

4.2.7 24-hour Cloud Cover Forecast

This section provides a visual comparison of the $AR-B-S-L5$-model, $ConvLSTM-B_{10}-SL_{24}-32-3\times3-32-3\times3$-model, ERA5 against ECC on their ability to forecast cloud

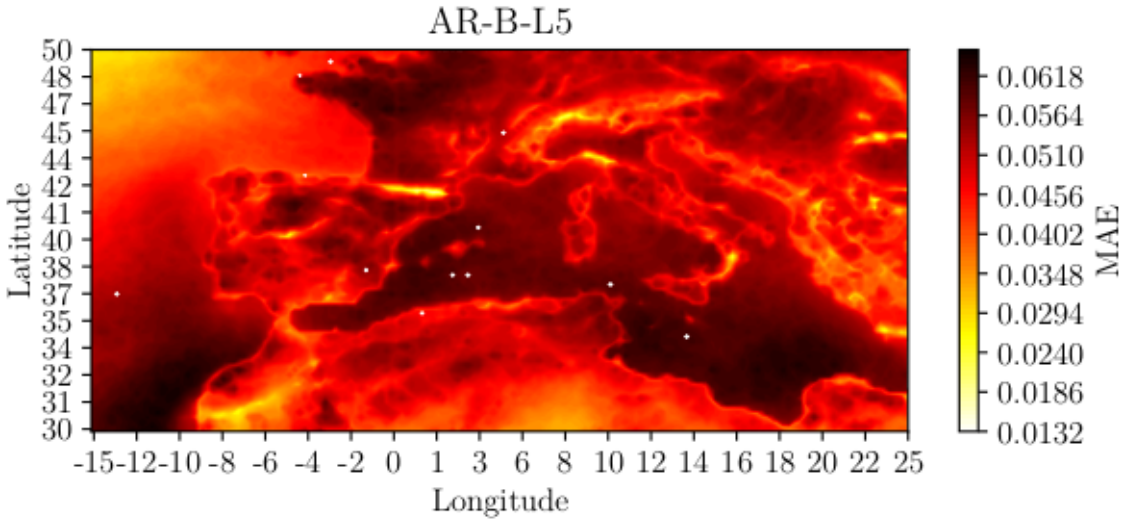


Figure 4.26: *Evaluation of the parameterization made by the $AR-B-L_5$-model. The white dots visible is in total 12/13041 regression model having numerical issues related to non-invertable matrices.*

	ERA5	**$AR-B-L_5$**	**$ConvLSTM-B_{10}-SL_{24}-32-3\times3-32-3\times3$**
MAE	0.204	0.485	0.336

Table 4.6: *MAE for the different cloud fractional cover parameterizations.*

cover for 24 hours starting from January 2, 2014.

To show the parameterizations side by side, the 24-hour forecast was split into four figures, each of six timesteps. The first one is shown here in Figure 4.27, and the full series consisting of Figures E.1 to E.4 can be found in Appendix E. The forecast produced by the different models are presented in their own column and the time progresses down the rows.

ERA5 resembles ECC the most. This is not surprising since it is the most complex parameterization included in this study, see Section 2.4.3 for more details. ERA5 distributes the cloud over the same regions as ECC, but it has smoother transitions between cloudy and non-cloudy areas, which leads to a reduced total cloudiness compared to ECC.

There are few clouds present in the first few hours of the ConvLSTM-produced forecast. The cloud fractional cover developed slowly. From $T09$ onwards, there is a significant overcast situation developed over parts of Europe and Egypt. In general the forecast is a lot blurrier

	ERA5	$\mathbf{AR-B-L_5}$	$\mathbf{ConvLSTM-B_{10}-SL_{24}-32-3\times3-32-3\times3}$
MAE	0.392	0.485	0.278
MIN	0.0	0.004	0.0
MAX	1.0	1.028	1.0

Table 4.7: *MAE, minimum and maximum values for the 24-hour forecast period of 2nd January 2014.*

than ECC. The time delay, present in the start of the forecast, may be explained by the fact that the forecast is not initiated with a cloud cover, only the environmental variables. A few timesteps may therefore be required in order to spin-up and fully develop the cloud cover.

At first sight, the $AR-B-L_5$ appears to be very sensitive to the initial conditions. The first hour is a remarkable match, while the following hours are not. It looks like the model creates a less cloudy copy of itself and presents this as the next prediction. Near the end of the forecast, almost all clouds have dissipated, leaving only a few in Europe. This "self-plagiarism" is most likely caused by the weights of L_1 being close to 0.9 as shown in Figure 4.20. The forecast is shown in isolation in Figure F.2 and the spatially averaged cloud cover is included in the title. In the start of the forecast is has a value of 0.67 which is reduced to 0.37 after 24-hours.

The forecast produced by $ConvLSTM-B_{10}-SL_{24}-32-3\times3-32-3\times3$-model contains no out of sample values, but it is worth mentioning that for the entire period (2014 to 2018) 0.5% of the values produced are below zero, the lowest being -1, while non are above 1. The minimum value in the forecast produced is positive, and the maximum is 1.028. Meaning its at some point, its 103% cloudy.

Table 4.7 shows the skill of the different models in predicting this 24 hours sequence. $ConvLSTM-B_{10}-SL_{24}-32-3\times3-32-3\times3$ ranks highest having a score of 0.204 , ERA5 ranks second with 0.33622 and $AR-B-L_5$ last with 0.48502. Again, not that a mean deviation of 0.46 is quite large when the data in question varies from 0 to 1.

To explore the importance of spatial information from neighboring pixels the $ConvLSTM-$

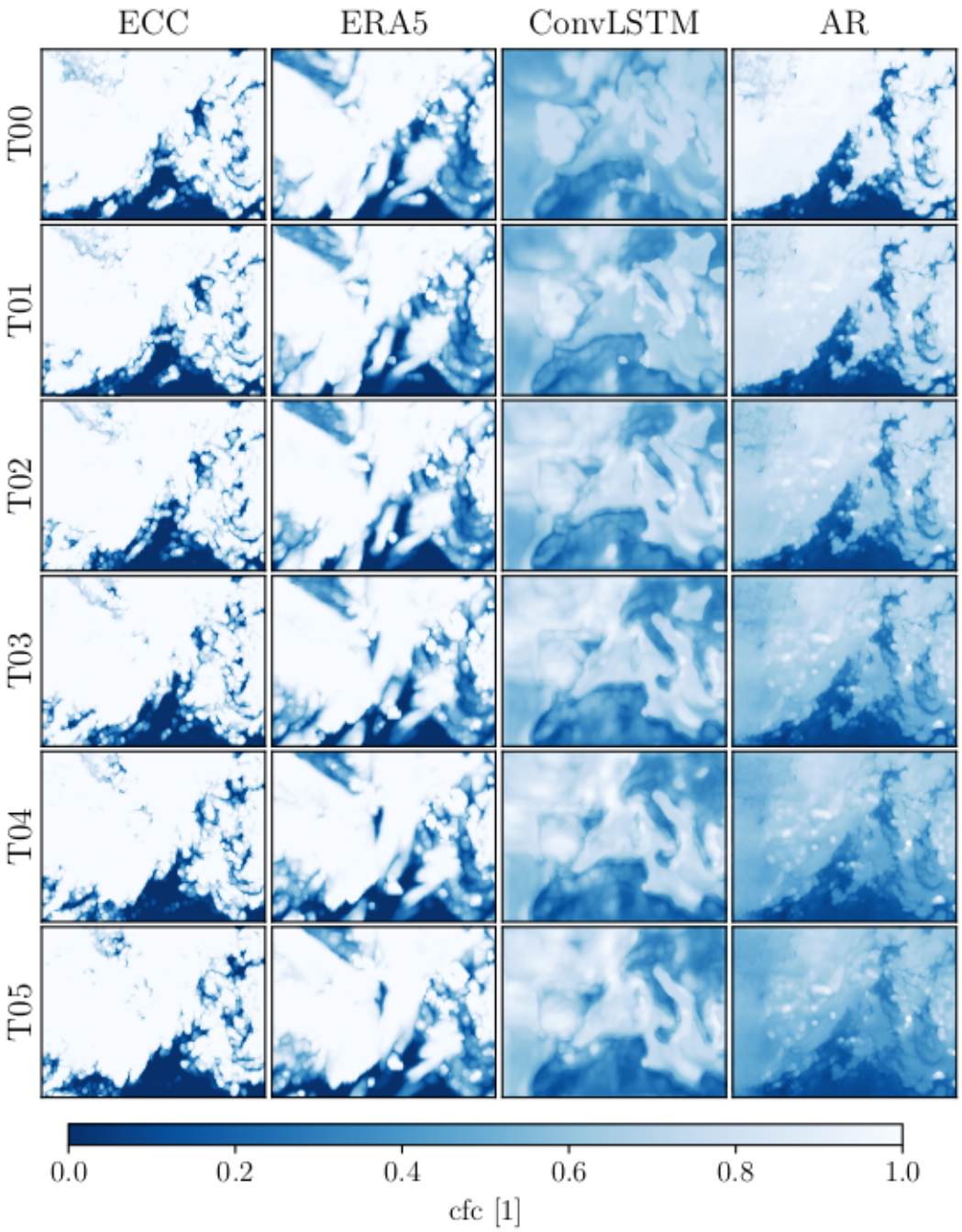


Figure 4.27: *First six hours of a 24-hours forecast. The full forecast can be found in Appendix E.*

$B_{10} - SL_{24} - 32 - 1 \times 1 - 32 - 1 \times 1$-model was trained and compared to the best model. As expected, this model has a higher loss than the architecture trained using a 3×3. This is shown in Table 4.5 summarizing the losses for all models in this study. Figure F.3 shows the

forecast produced using the $1 \times 1 - filter$. Contrary to what the higher loss suggest, i.e. bad performance, the model appears to produce a more realistic forecast than $ConvLSTM - B_{10} - SL_{24} - 32 - 3 \times 3 - 32 - 3 \times 3$, which includes information from neighboring pixels.

4.2.8 Practical implications

A major downside of the data driven learning approach is the rigid resolution. A trained model can only be used on similar problems, with the same spatiotemporal resolution. For applications like climate models, output comes in a wide range of different resolutions. Before implementing the finished product in a new model of a different resolution, it would need to be retrained on the resolution of the climate model under development. This process involves both remapping of the dataset and retraining the model at the correct resolution. This is a time consuming process involving finding a new set of hyperparameters suitable for the new resolution.

Once trained on global climate datasets, machine learning models provide fast results even for complex parameterization, which is what makes them suitable for the application of climate modelling. Most machine learning packages are developed using Python. Earth System Model (ESM) are increasingly also implemented in python. Methods for including the trained parameterizations need to be developed to further explore the potential of machine learning for climate modeling in the future.

5 Conclusions

5.1 Summary and main contributions

This study has contributed with research in a popular field, which is applying data driven learning in the form of DL to perform climate predictions. This is a "proof of concept"-study, exploring the potential for data driven learning in applications to parameterization of clouds. The developed methodology is transferable to other similar issues in Geosciences.

In this study a high quality dataset, European Cloud Cover (ECC) was compiled for the purpose of performing data driven learning on cloud fractional cover. This involves the development of Area Weighting Regridding Scheme (AWRS) and derivation of relevant equations. Without this dataset proposed methods would be useless. The data basis, both the dataset and variables, has been carefully chosen. It consists of temperature, pressure, relative and specific humidity from ERA5 and cloud fractional cover, computed using AWRS, based on cloud masks from MSG. The final product has 0.25^o spatial and hourly resolution.

Two models, Autoregressive (AR) and Convolutional Long Short-Term Memory (ConvLSTM) models were built on ECC. They were evaluated on their abilities to produce a 24-hour forecast and compared against existing parameterizations in ERA5.

$AR-B-L_5$ sufferers from "self-plagiarism" when it produces a forecast. Each predictions is a less cloudy version of the previous one. Consequently, it is unfit to produce a realistic cloud cover forecast. The $ConvLSTM-B_{10}-SL_{24}-32-3\times3-32-3\times3$-model showed promise and a few hours into the forecast the distribution of clouds started to bear a resemblance to the cloud cover in ECC. However, this type of model shows issues with blurred lines. The $ConvLSTM-B_{10}-SL_{24}-32-1\times1-32-1\times1$ had a higher loss, but has a superior performance when producing the 24-hour forecast.

This study has successfully applied data driven learning to the cloud forecasting problem. An important and valuable contribution was made by building an end-to-end trainable ConvLSTM-model based on ECC, training a ConvLSTM-model with a larger dataset than earlier studies. This is a indication that there may in fact be sufficient information in simple meteo-

rological variables to perform climate predictions.

In conclusion, at their current state data driven learning approaches are not appropriate for implementations toward practical applications. However, they show potential for further studies and development, as this proof of concept study was verified.

5.2 Future work

To asses the numerical issues related to the AR-models, one approach worth investigating is applying regularization, which may also influence performance. To improve upon the AR-models' ability to predict sequences it would be interesting to train different AR-model for each timestep in a sequence, using the same input data but training one model to fit the first hour and another one to fit second, third and so on.

Cloud cover varies in the range between 0 and 1. One way of guaranteeing that the models sticks to this range could be to employ the sigmoid function. For the ConvLSTM-model this would be implemented as the activation function of the output layer. In the case of AR-models it means fitting the regression models against an inversely transformed target, to then transform the prediction back.

ConvLSTM-models provide an extensive list of hyperparameters and this study has not explored the full range of possibilities they potentially enable. Future studies should employ automatic hyperparameter optimization, in combination with the limits of hyperparameters tuned in this study.

Weather patterns and the number of daylight hours determine the seasons. Cloud cover is heavily affected by both and it would be interesting to evaluate the parameterization's ability to capture the cloud fractional cover for varying seasons.

A reasonably accurate model can be useful in practical applications. It would be interesting to evaluate the improved parameterizations in the context of a full climate model. This can be done using the output from the ensemble simulations in e.g. CMIP6. This could answer important questions like whether the developed parameterizations generalizable to a warmer climate.

A Statistics

A.1 Temporal Statistics and Deviation

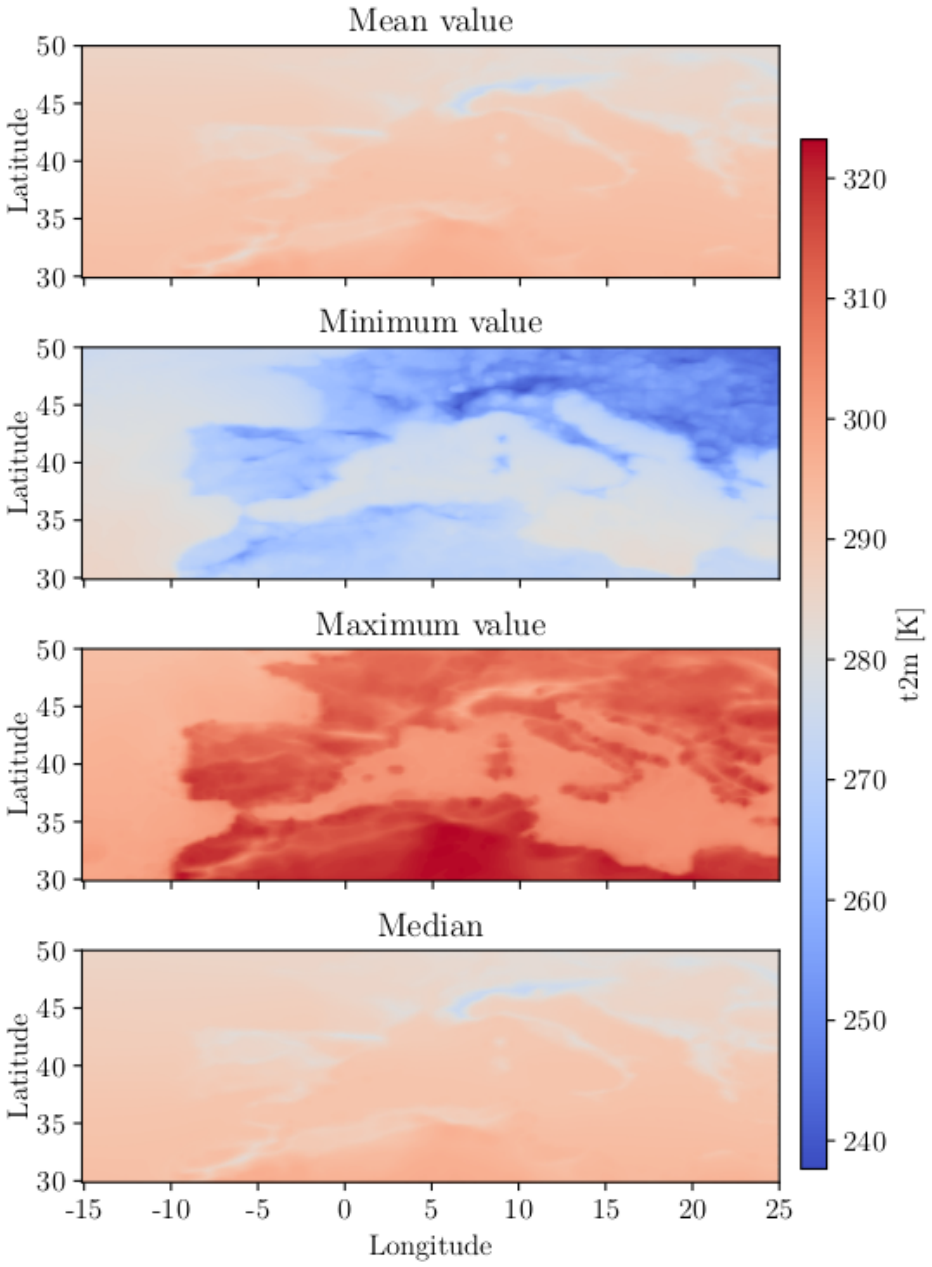


Figure A.1: *Contour plot showing the local (pixel) statistics for temperature.*

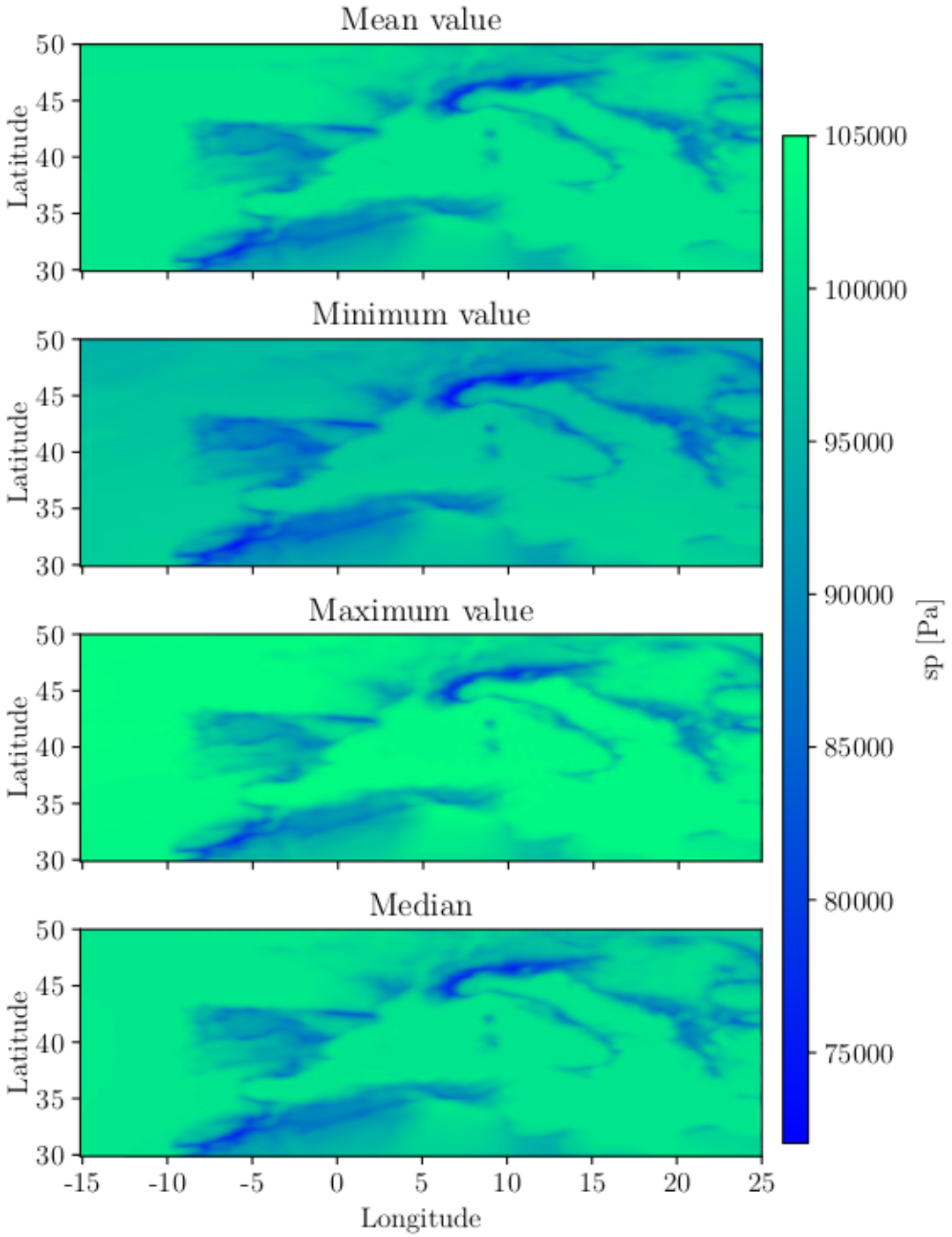


Figure A.2: *Contour plot showing the local (pixel) statistics for surface pressure.*

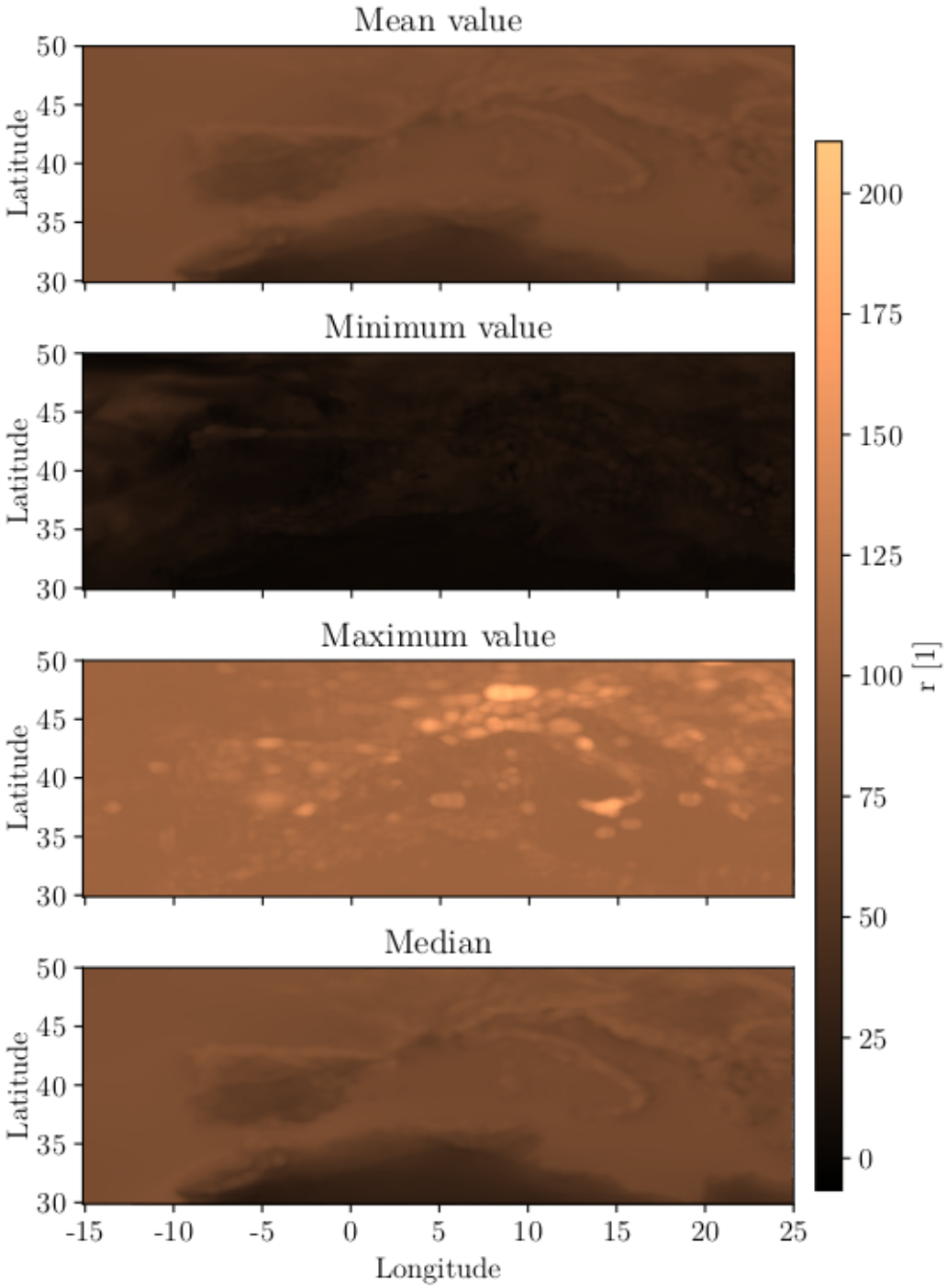


Figure A.3: *Contour plot showing the local (pixel) statistics for relative humidity.*

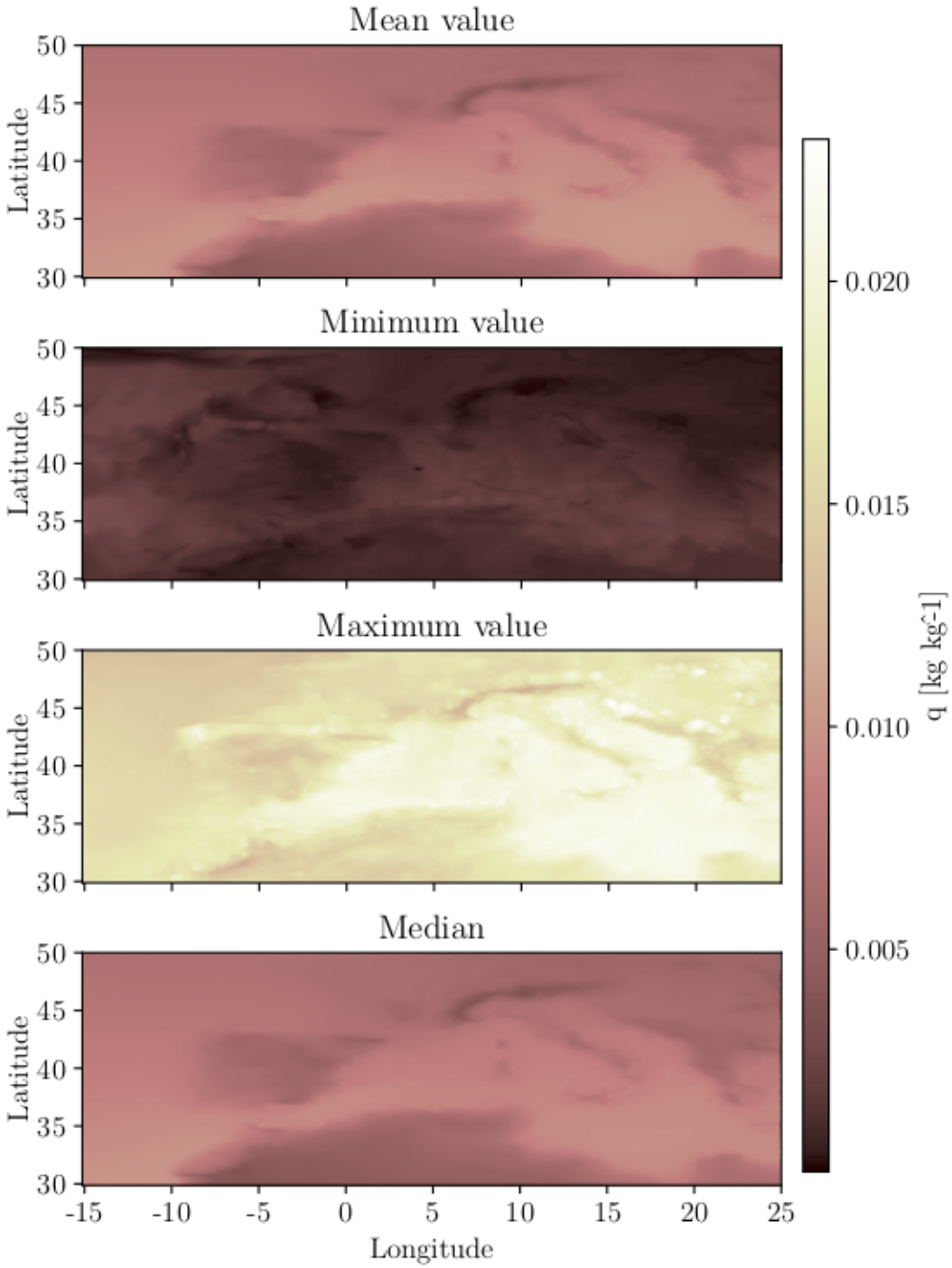


Figure A.4: *Contour plot showing the local (pixel) statistics for specific humidity.*

A.2 Temporal Statistics Deviations

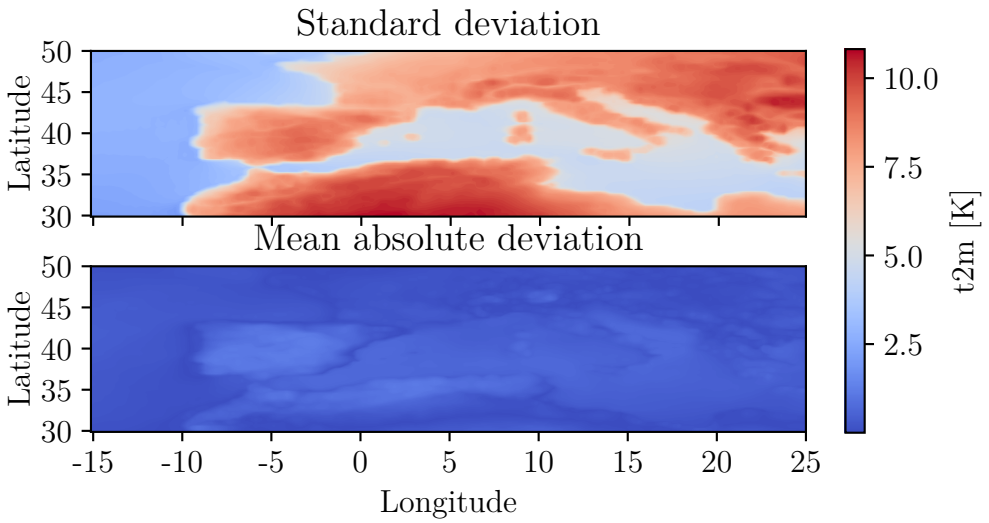


Figure A.5: *Deviations in two meter temperature, t_2m.*

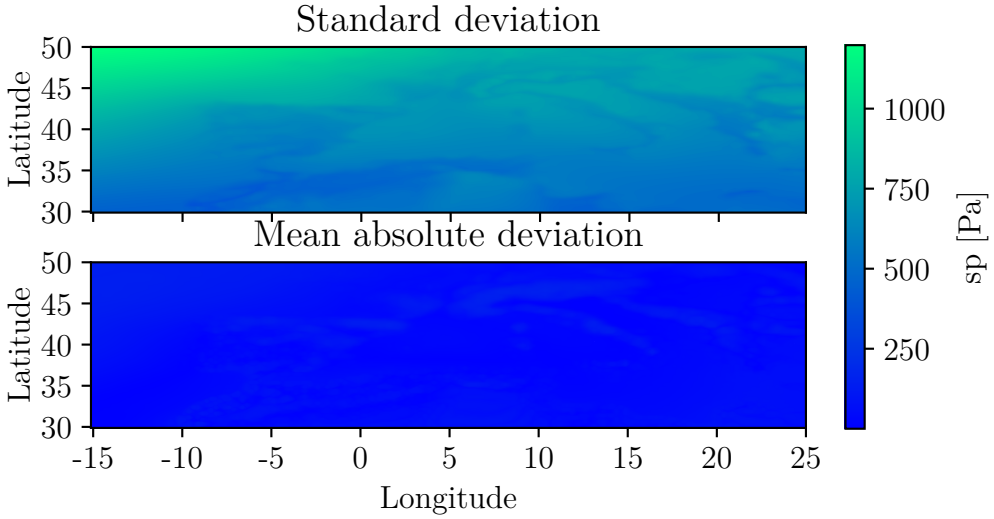


Figure A.6: *Deviations in surface pressure, sp.*

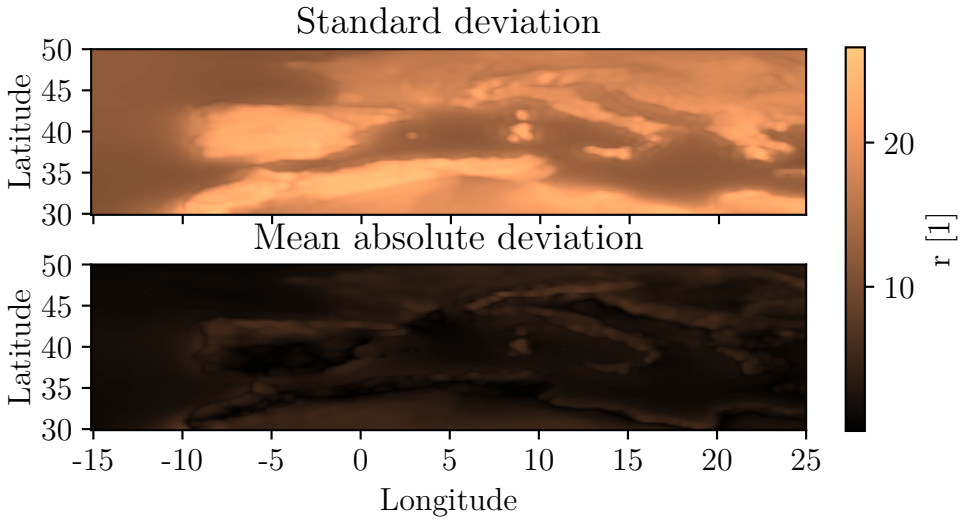


Figure A.7: *Deviations in relative humidity, r.*

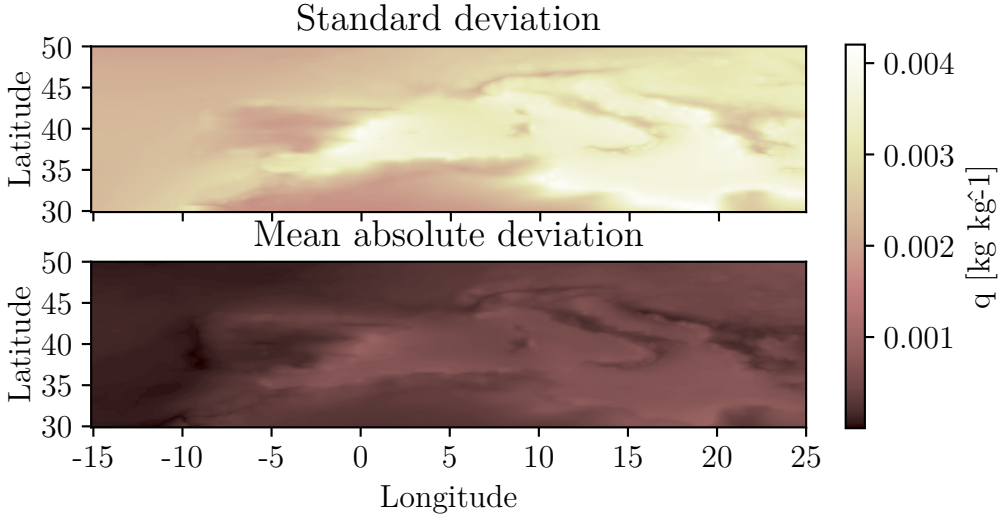


Figure A.8: *Deviations in specific humidity, q.*

B Seasonal effects

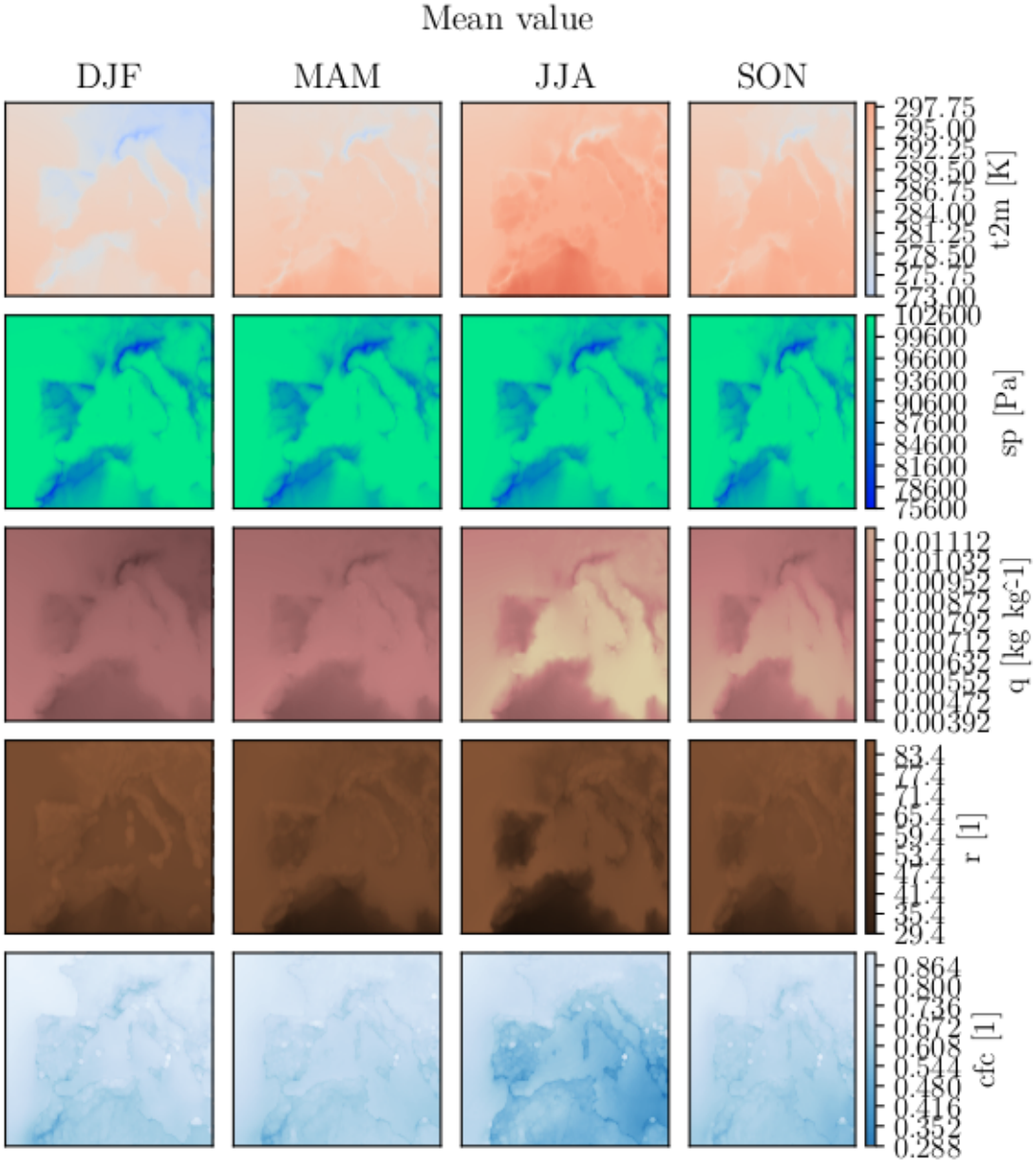


Figure B.1: *Seasonal mean for all variables in ECC.*

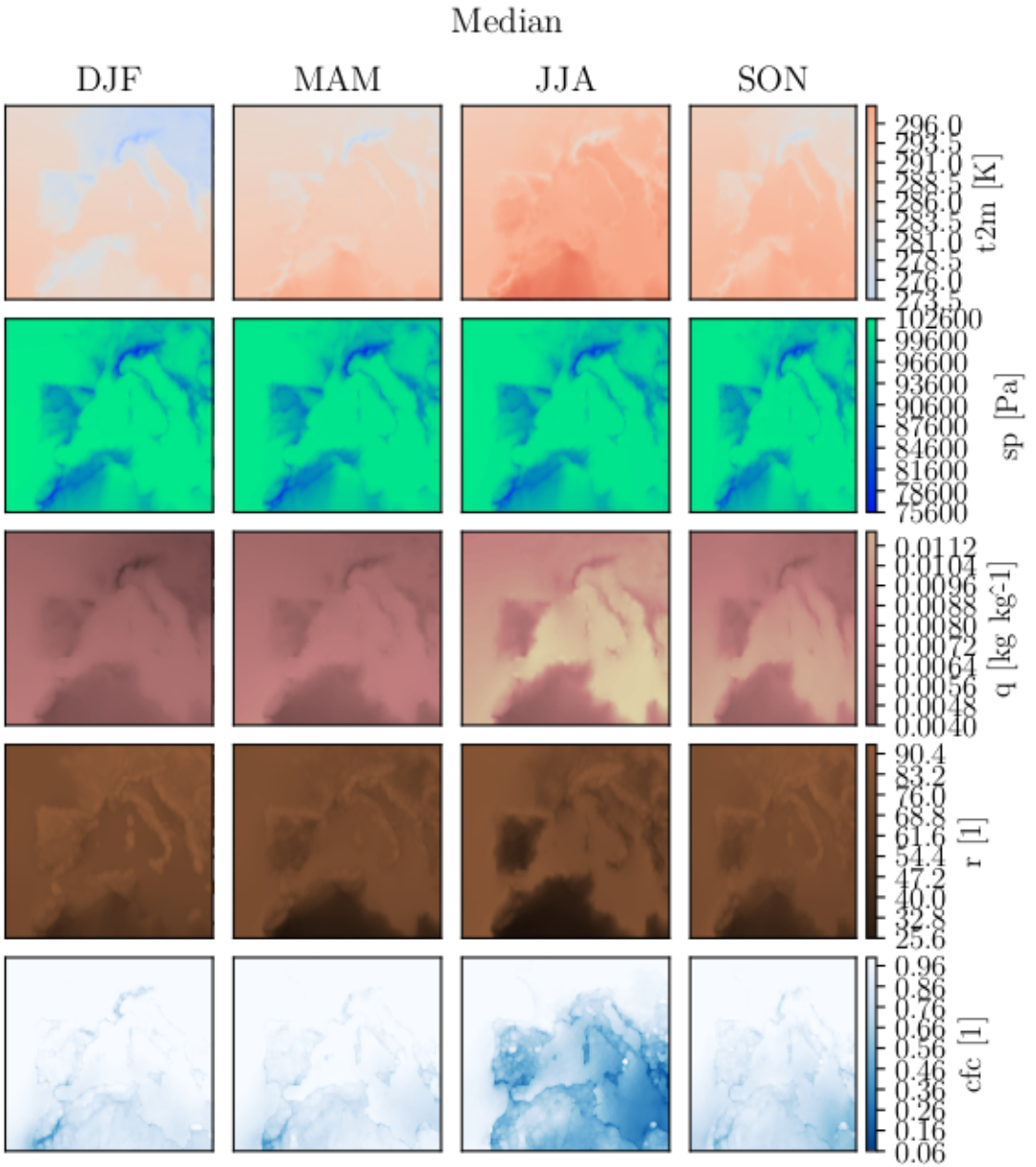


Figure B.2: *Seasonal median for all variables in ECC.*

C Series of First Weeks in 2012

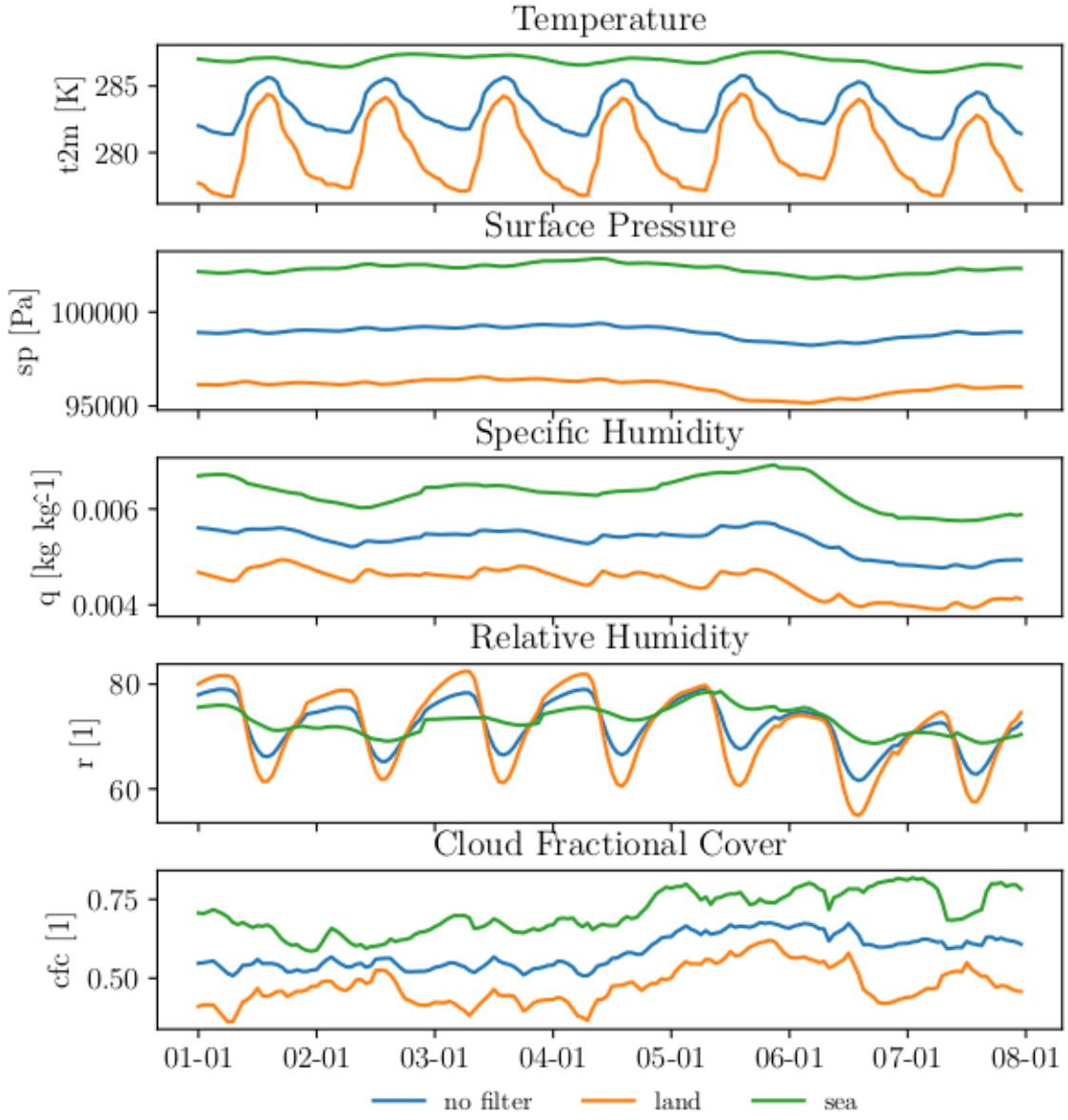


Figure C.1: *ECC variables first week January 2012.*

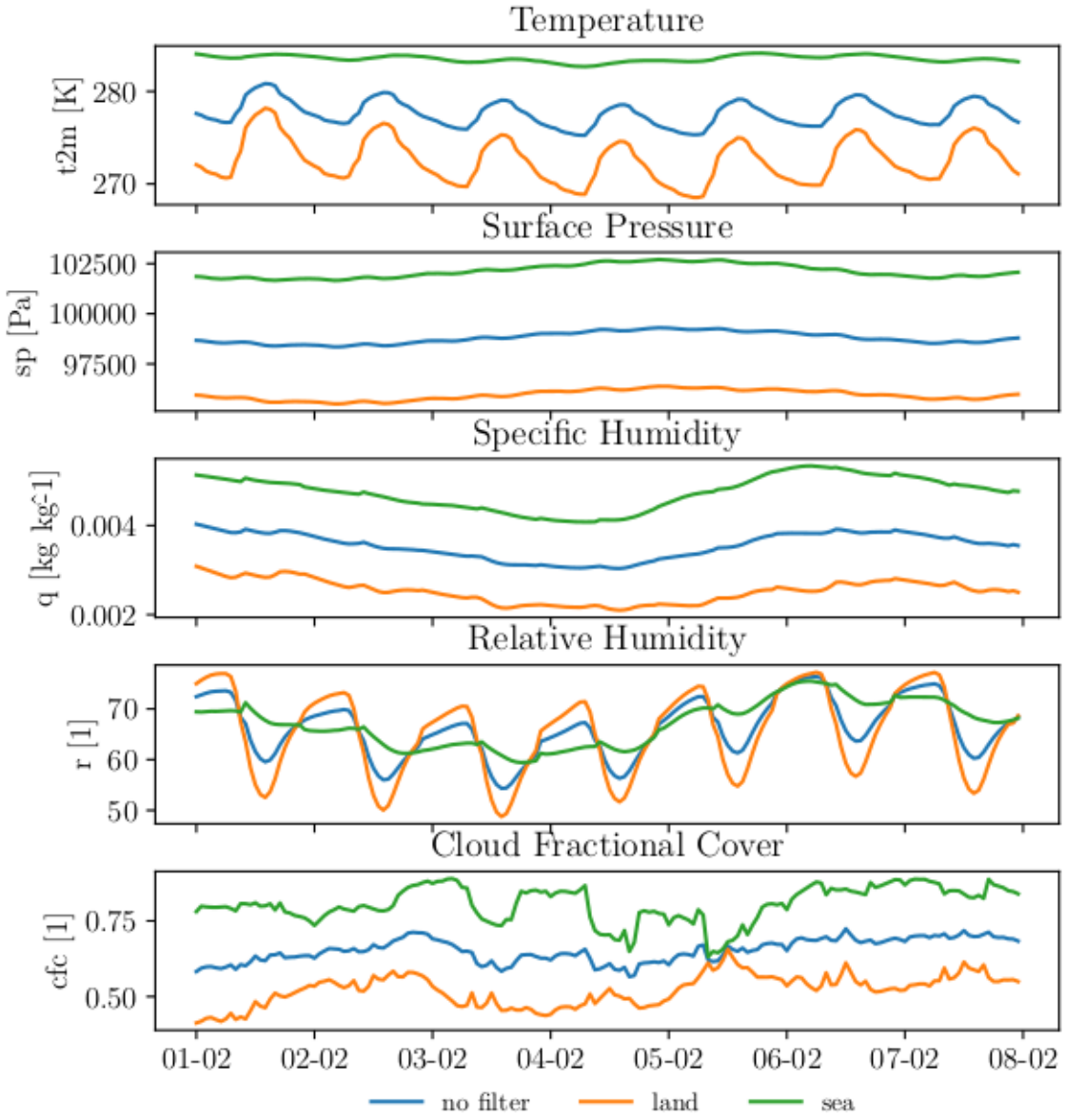


Figure C.2: *ECC variables first week February 2012.*

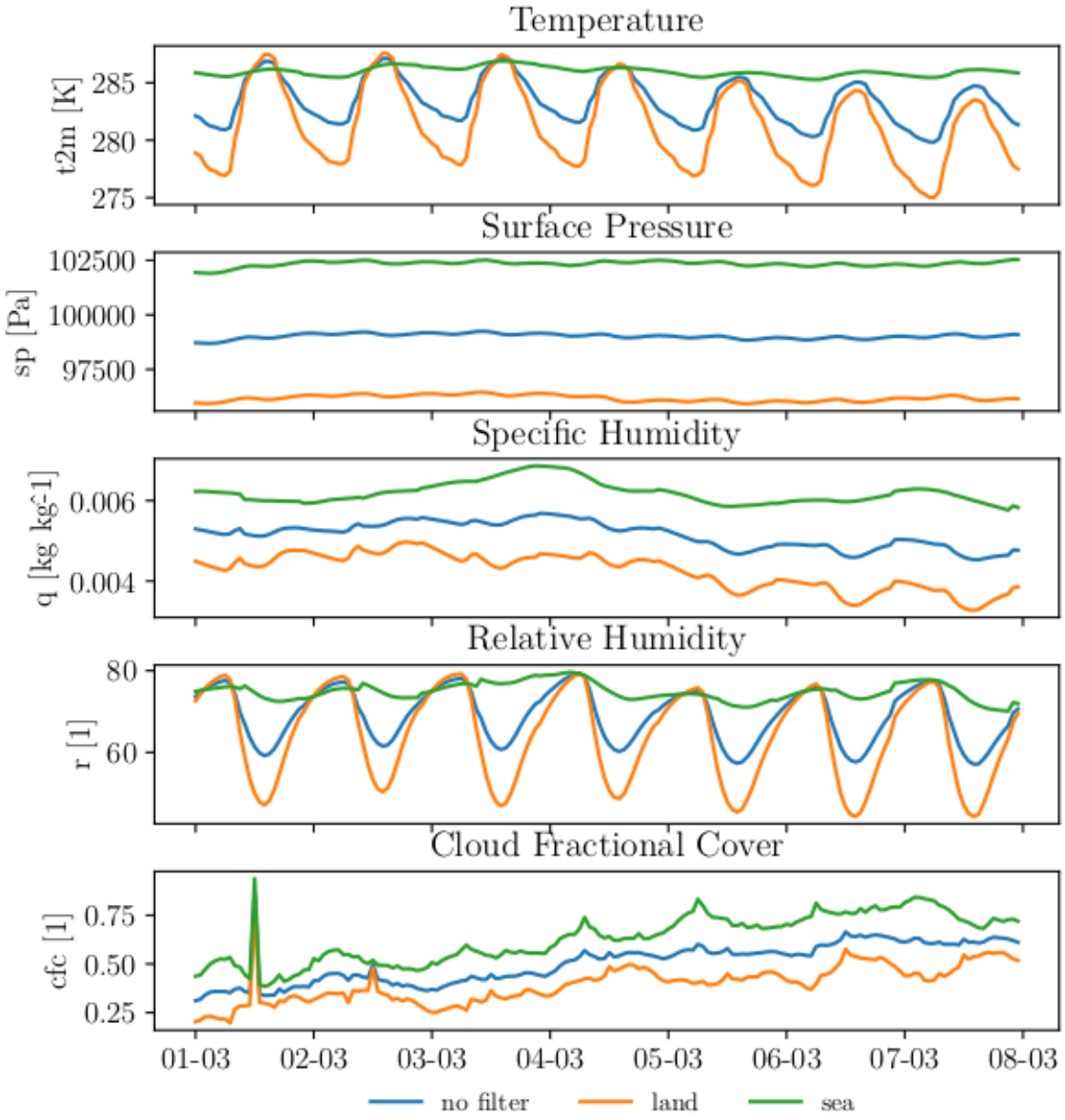


Figure C.3: *ECC variables first week March 2012.*

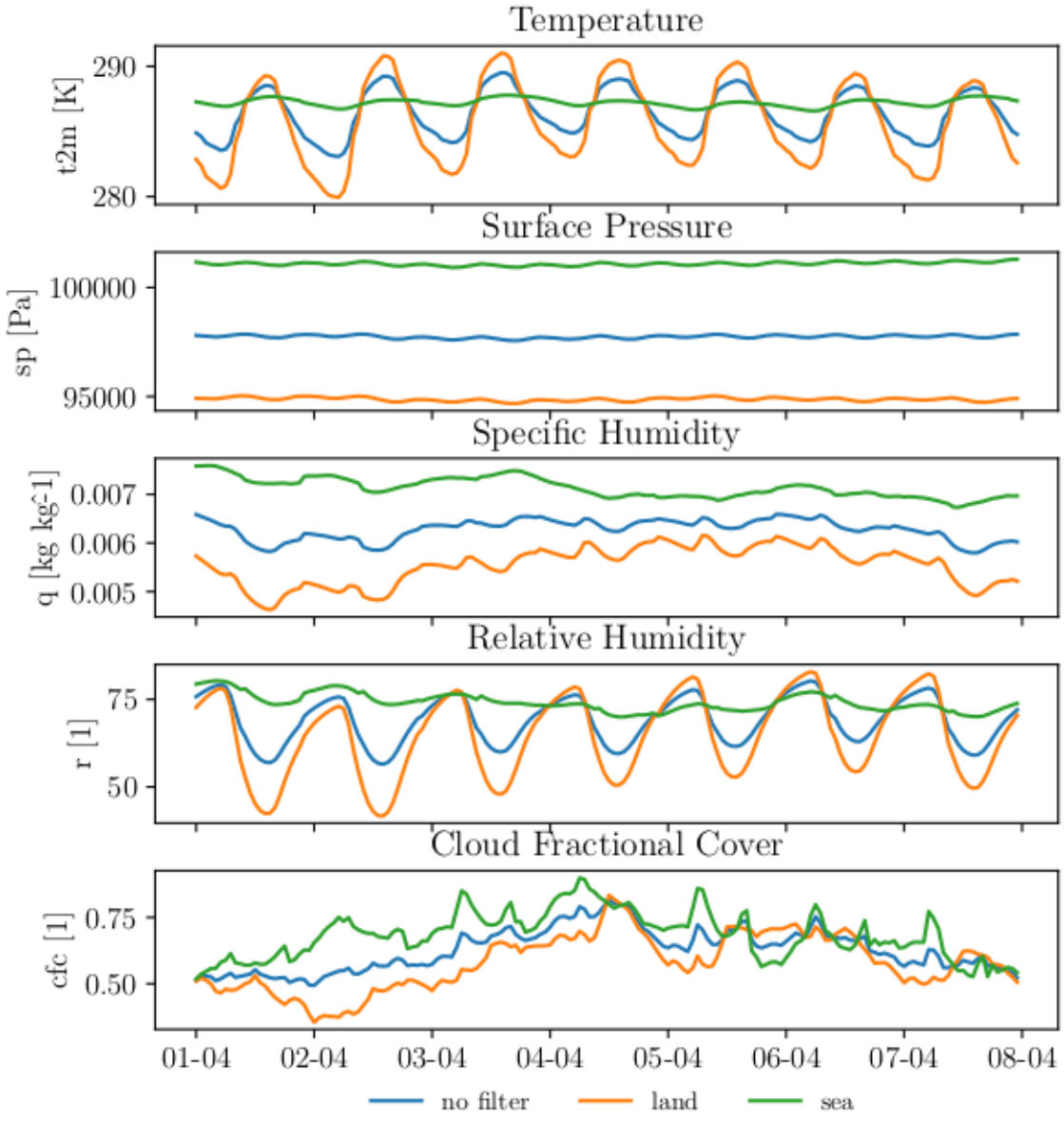


Figure C.4: *ECC variables first week April 2012.*

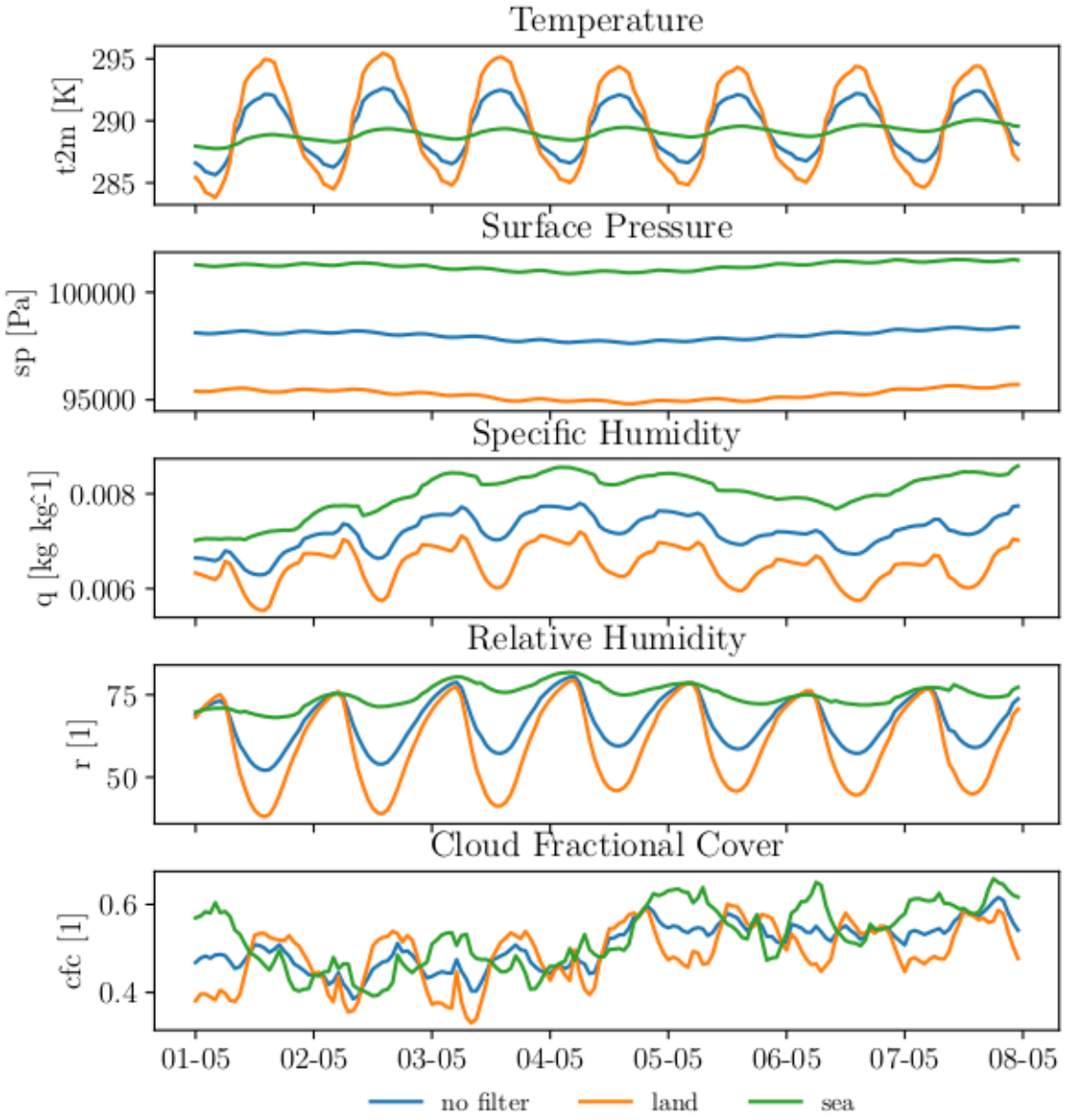


Figure C.5: *ECC variables first week May 2012.*

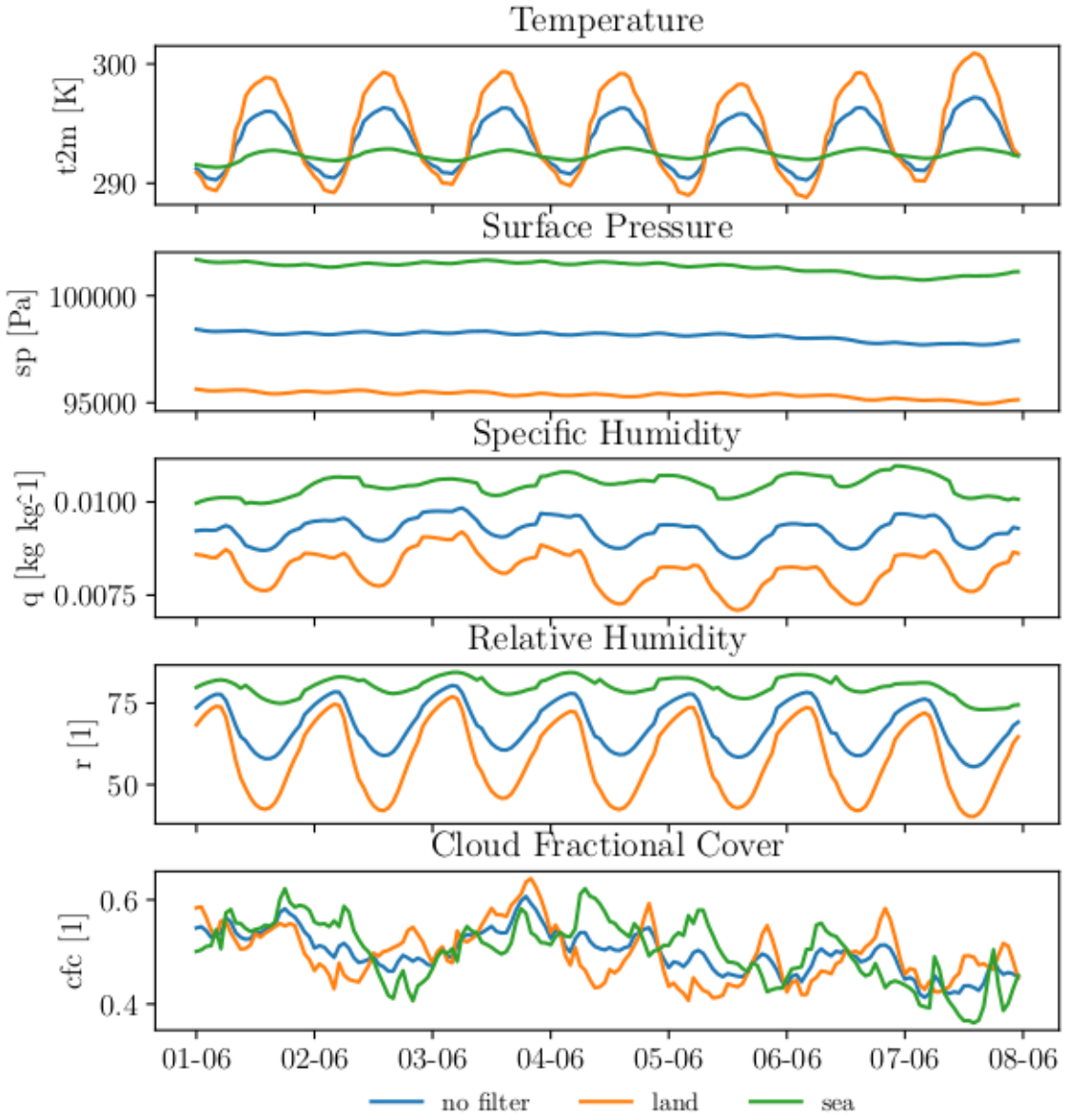


Figure C.6: *ECC variables first week June 2012.*

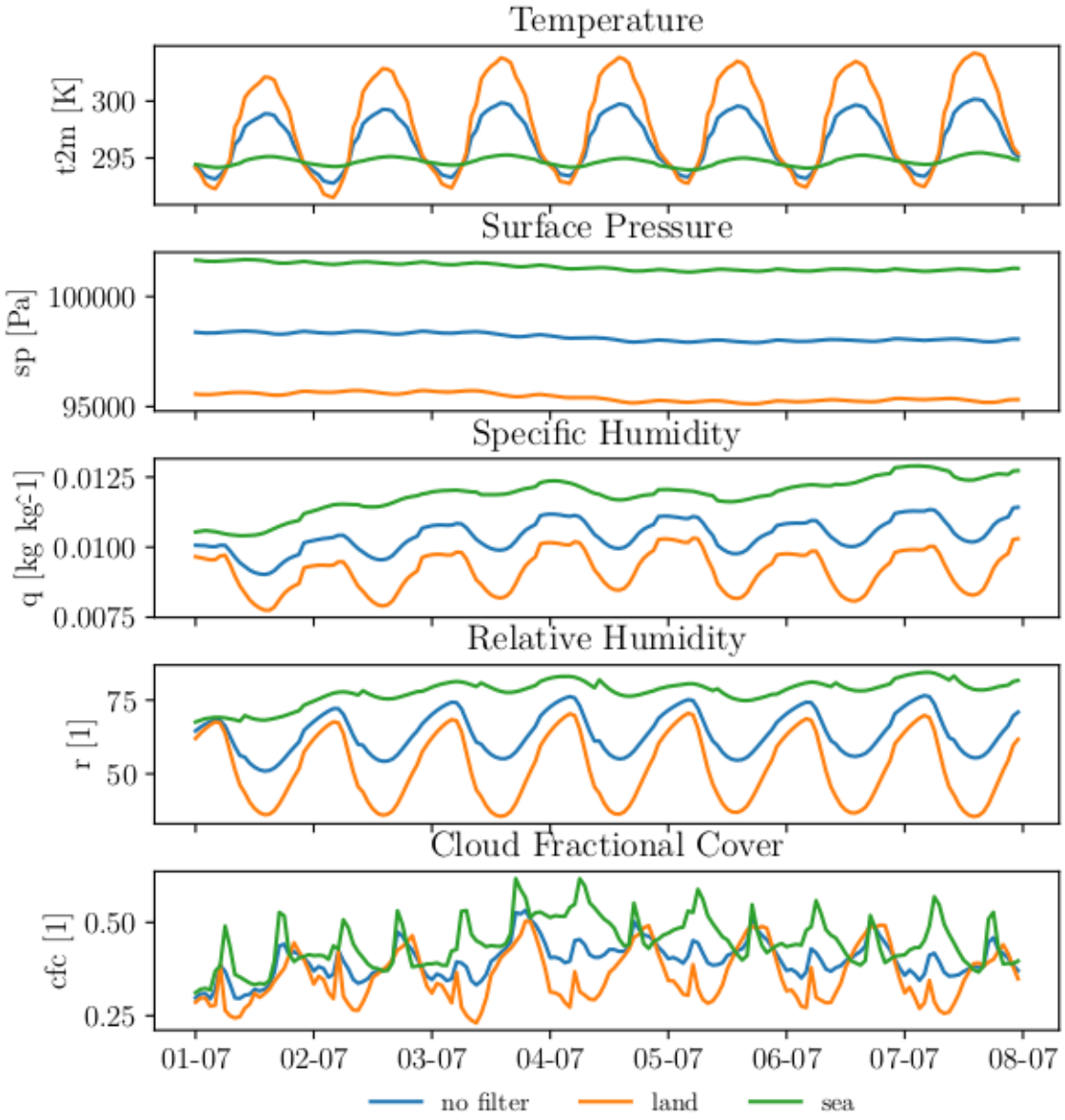


Figure C.7: *ECC variables first week July 2012.*

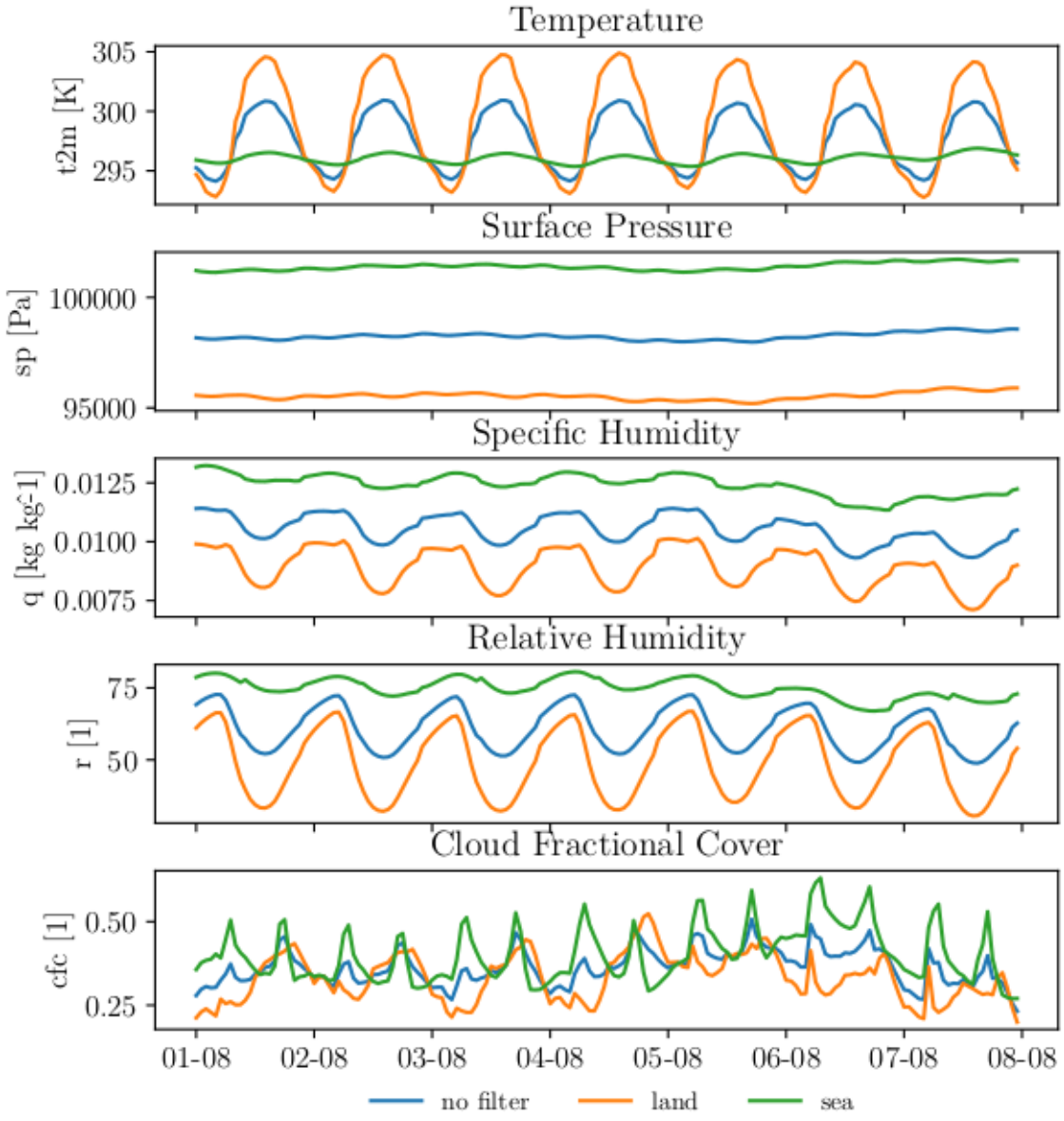


Figure C.8: *ECC variables first week August 2012.*

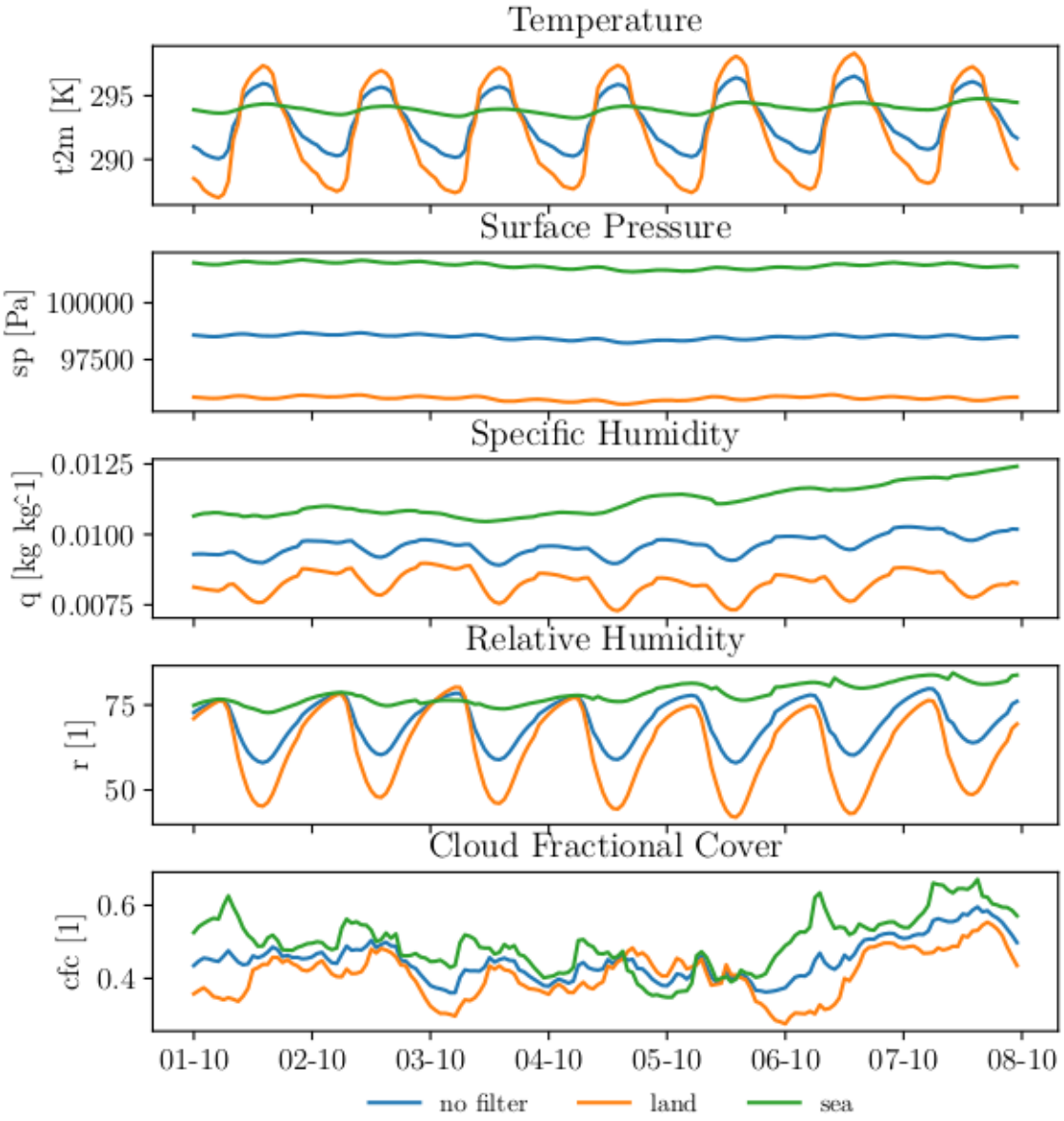


Figure C.9: *ECC variables first week October 2012.*

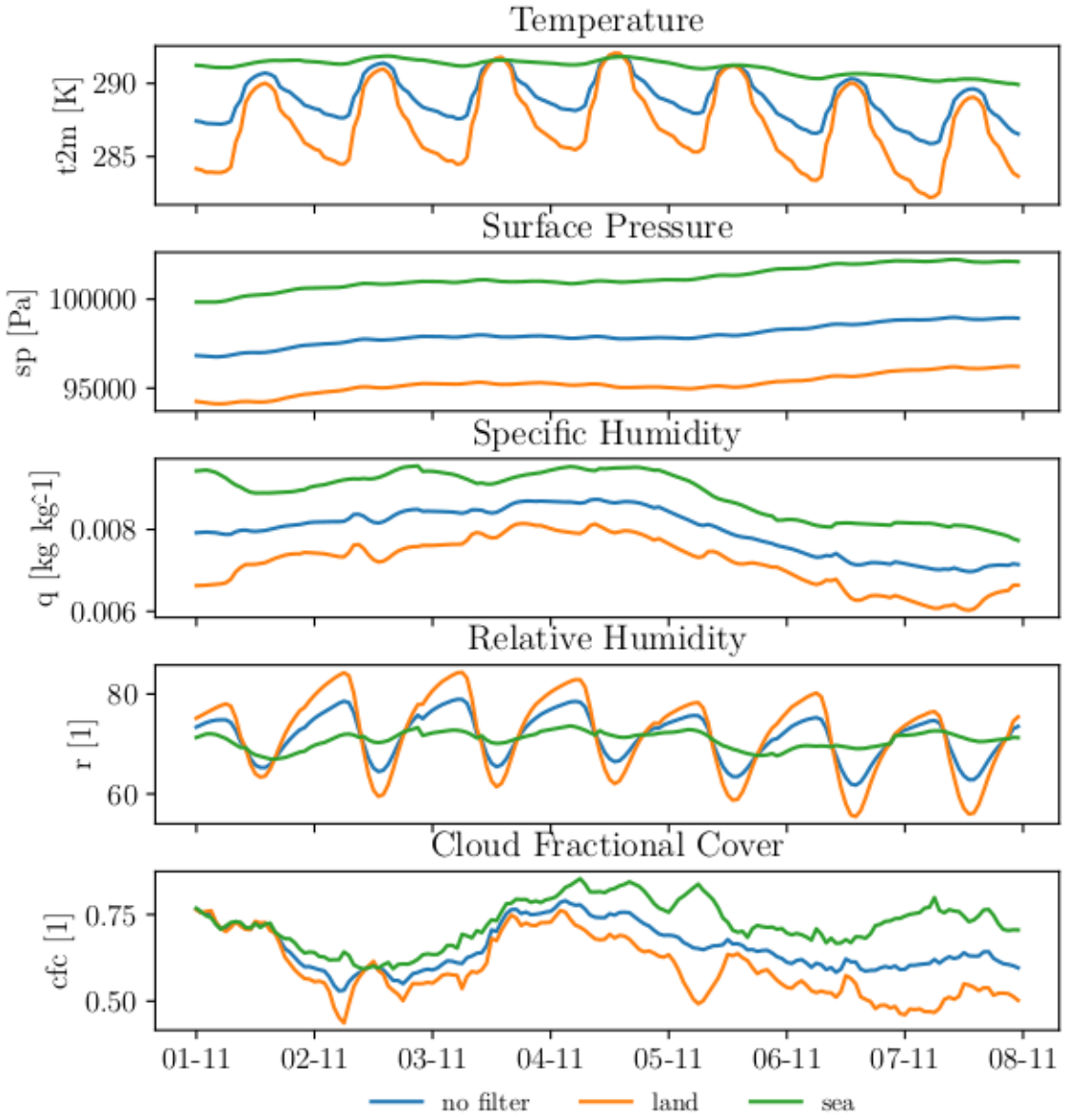


Figure C.10: *ECC variables first week November 2012.*

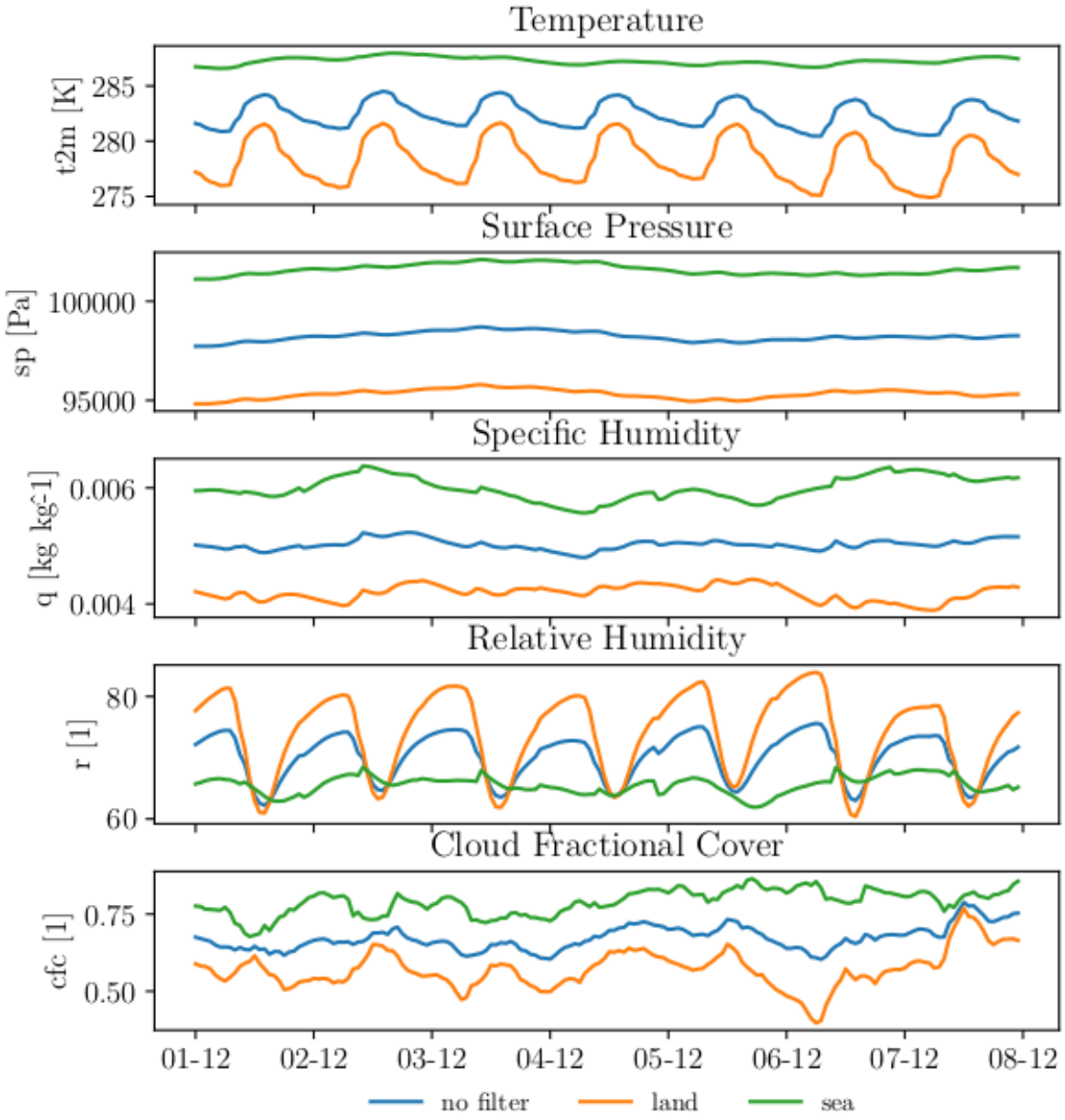


Figure C.11: *ECC variables first week December 2012.*

D List of non-trainable architectures

1. $ConvLSTM - B_{10} - SL_{24} - 128 - 3 \times 3$
2. $ConvLSTM - B_{10} - SL_{24} - 128 - 3 \times 3 - 32 - 3 \times 3$
3. $ConvLSTM - B_{10} - SL_{24} - 64 - 3 \times 3 - 64 - 3 \times 3$
4. $ConvLSTM - B_{5} - SL_{24} - 256 - 3 \times 3 - 256 - 3 \times 3$
5. $ConvLSTM - B_{5} - SL_{6} - 256 - 3 \times 3 - 256 - 3 \times 3$

D.1 Performance of AR-models

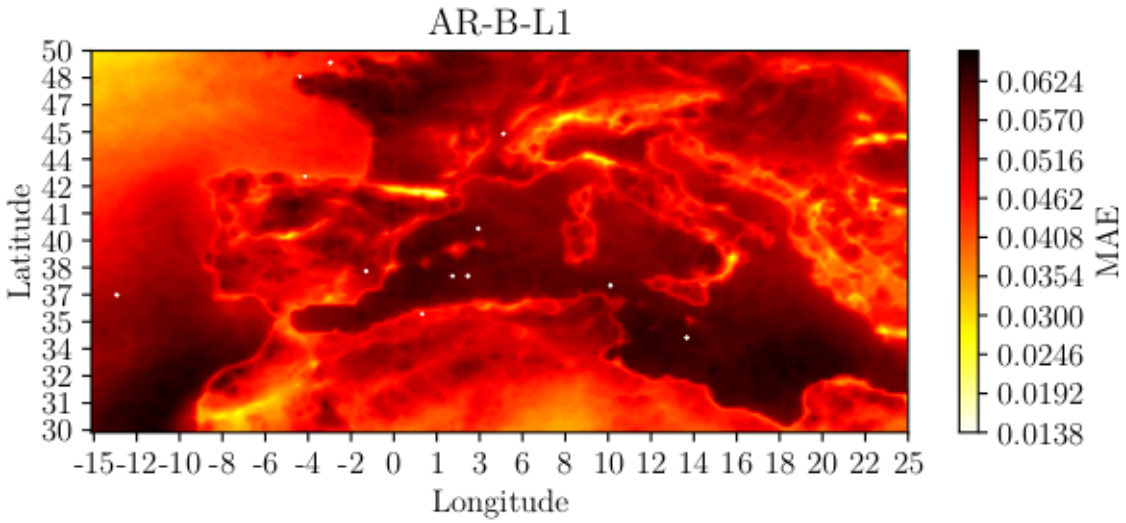


Figure D.1: *MAE of model* $AR-B-L_1$.

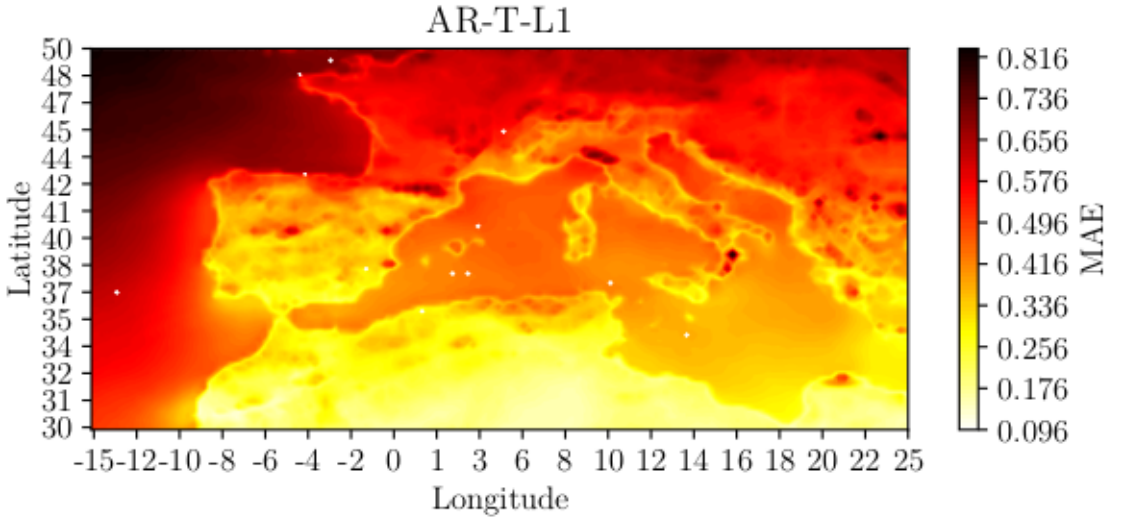


Figure D.2: *MAE of model* $AR-T-L_1$.

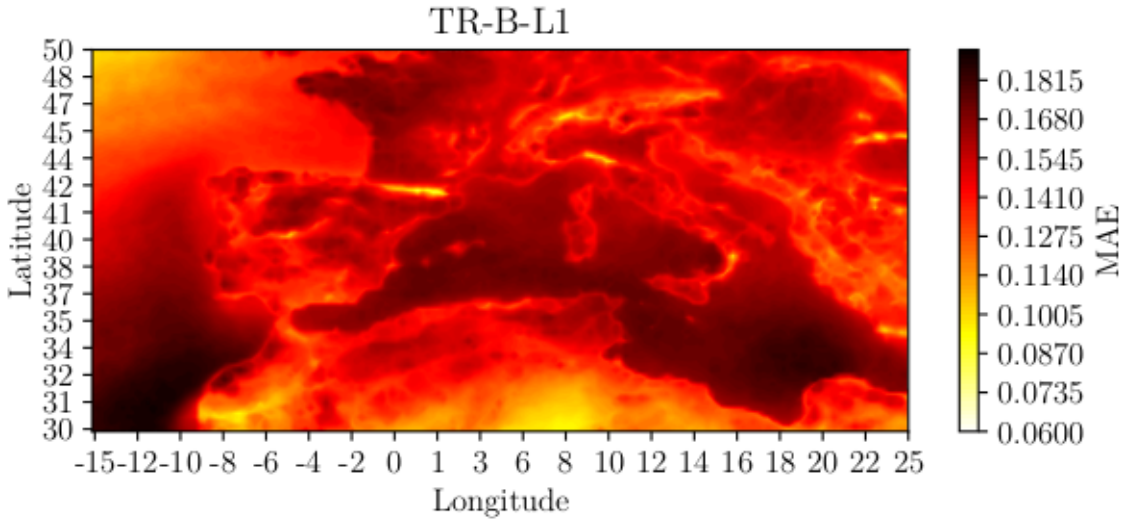


Figure D.3: *MAE of model* $TR-B-L_1$.

E Series of 24-hour Forecast

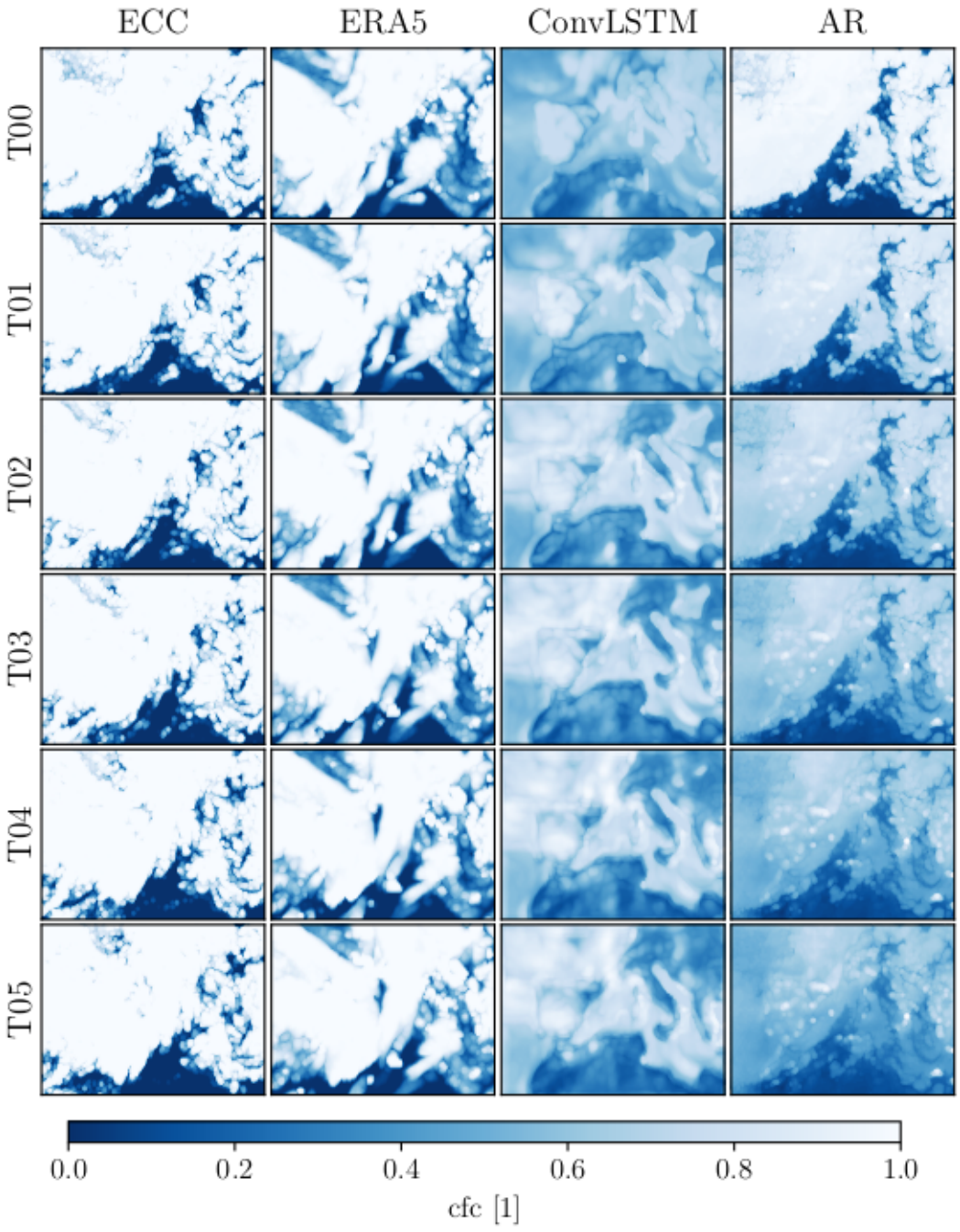


Figure E.1: *Hours zero to five of the forecast.*

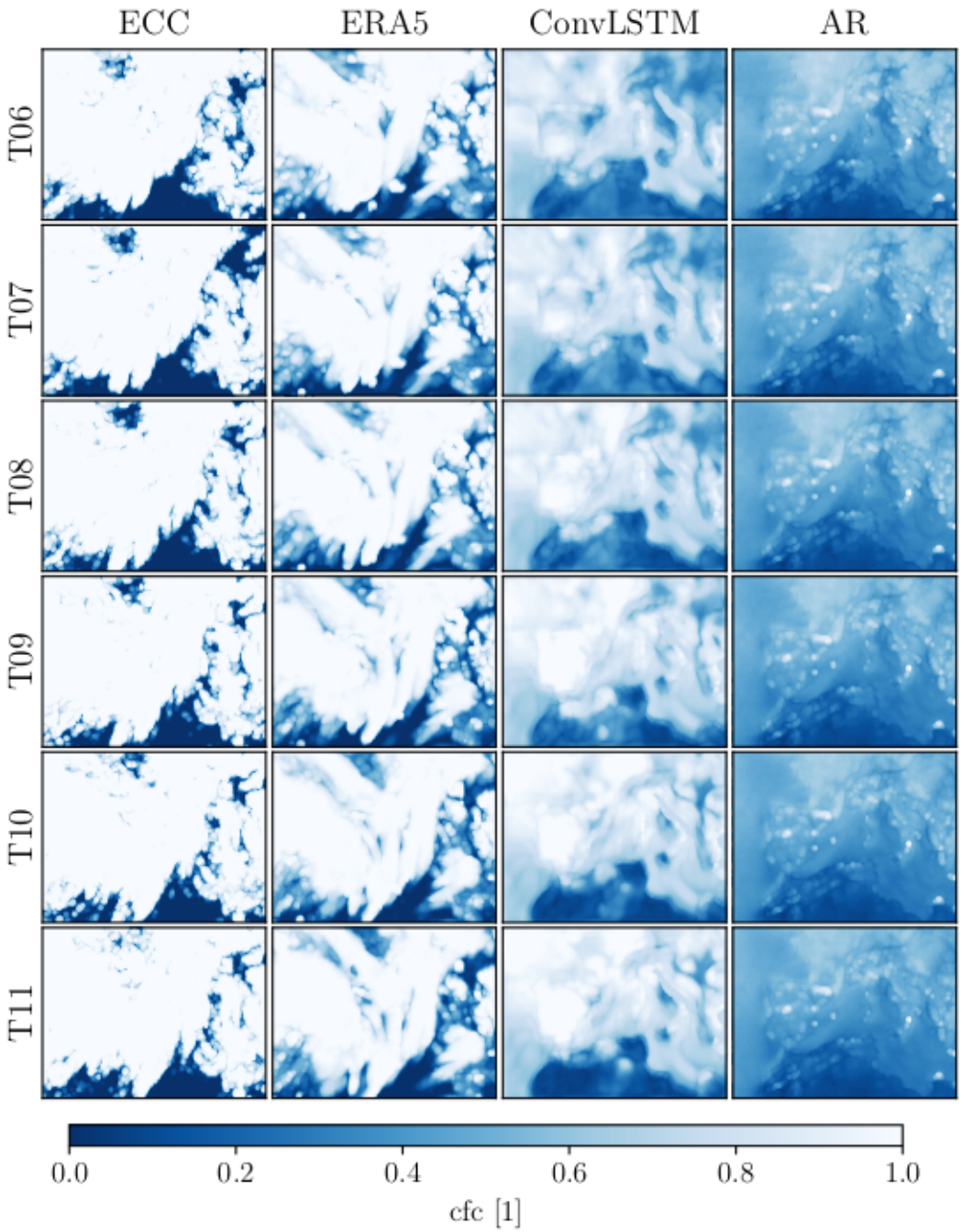


Figure E.2: *Hours six to 11 of the forecast.*

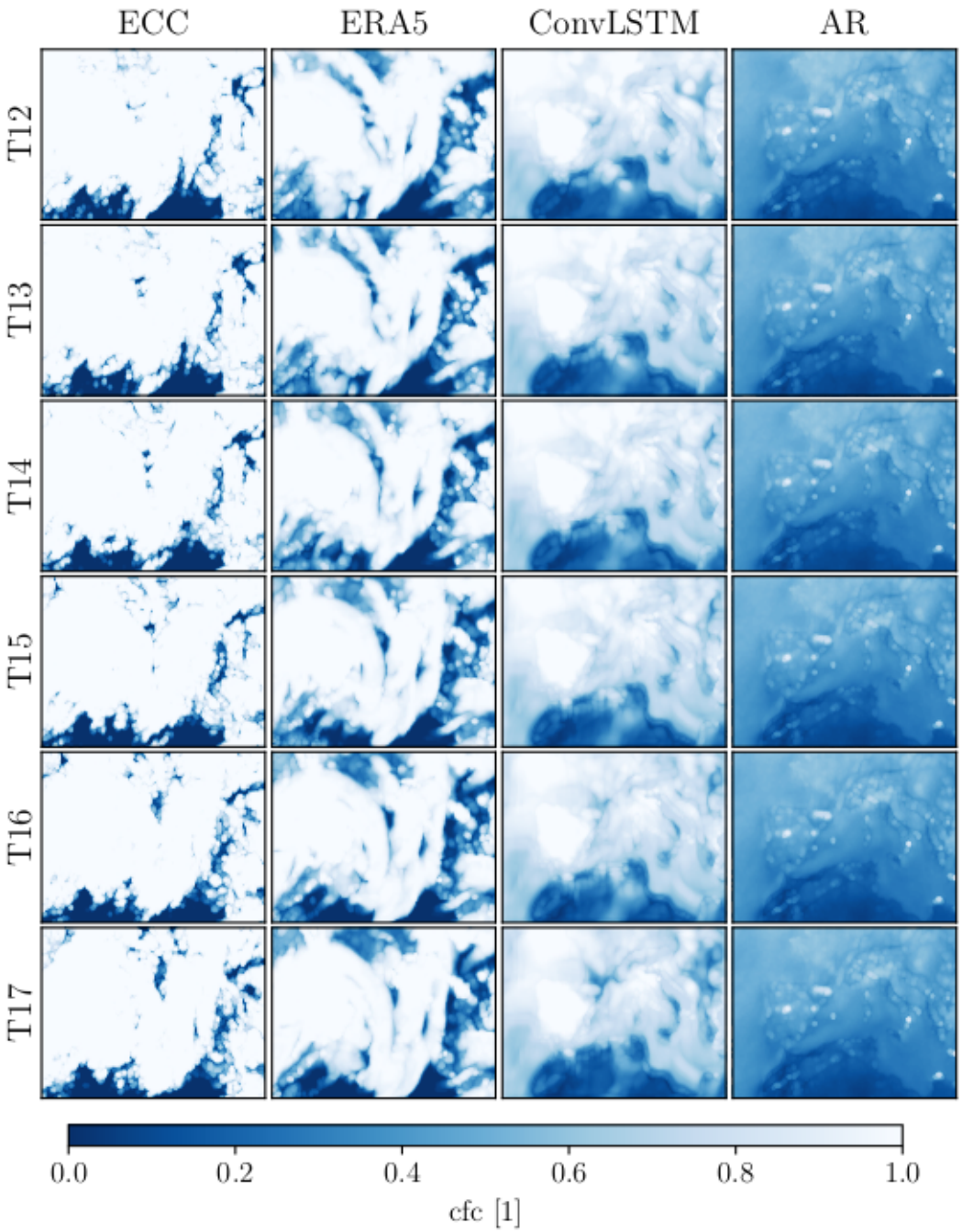


Figure E.3: *Hours 12 to 17 of the forecast.*

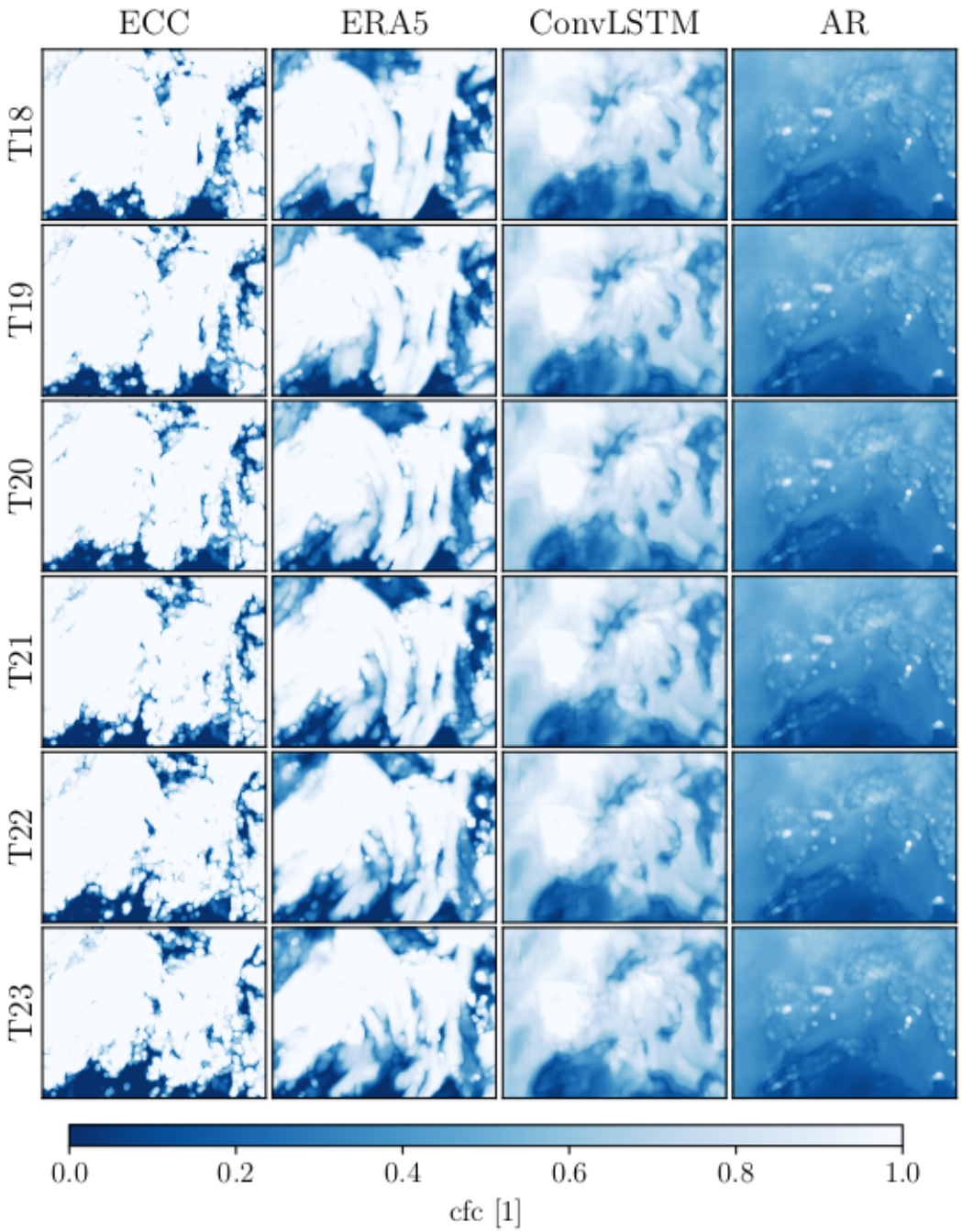


Figure E.4: *Hours 18 to 23 of the forecast.*

F 24-hour Forecasts

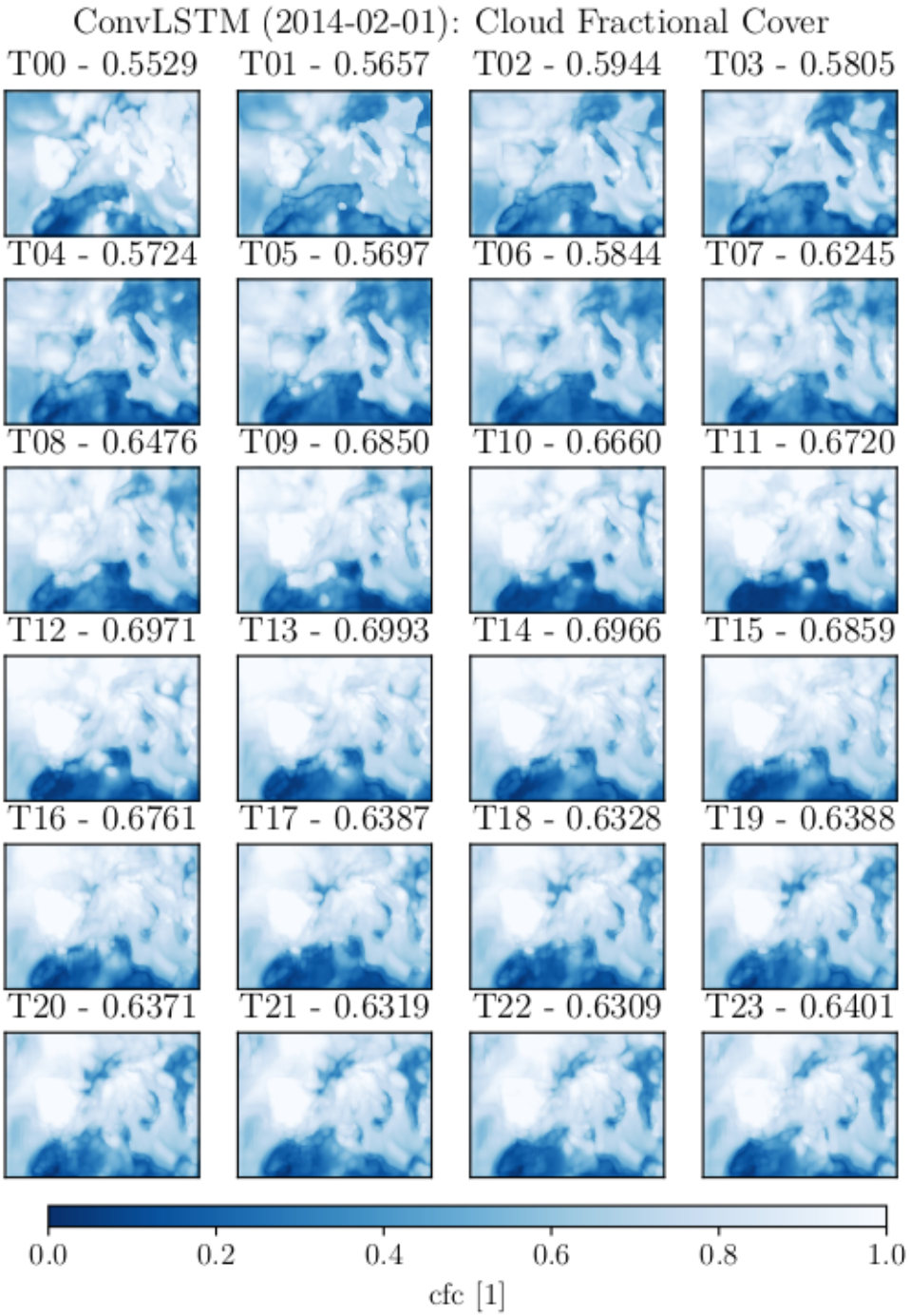


Figure F.1: *Cloud cover forecast produced by* $ConvLSTM - B_{10} - SL_{24} - 32 - 3 \times 3 - 32 - 3 \times 3$. *Initiated on January 2, 2014.The area mean cloud fraction is included in the title.*

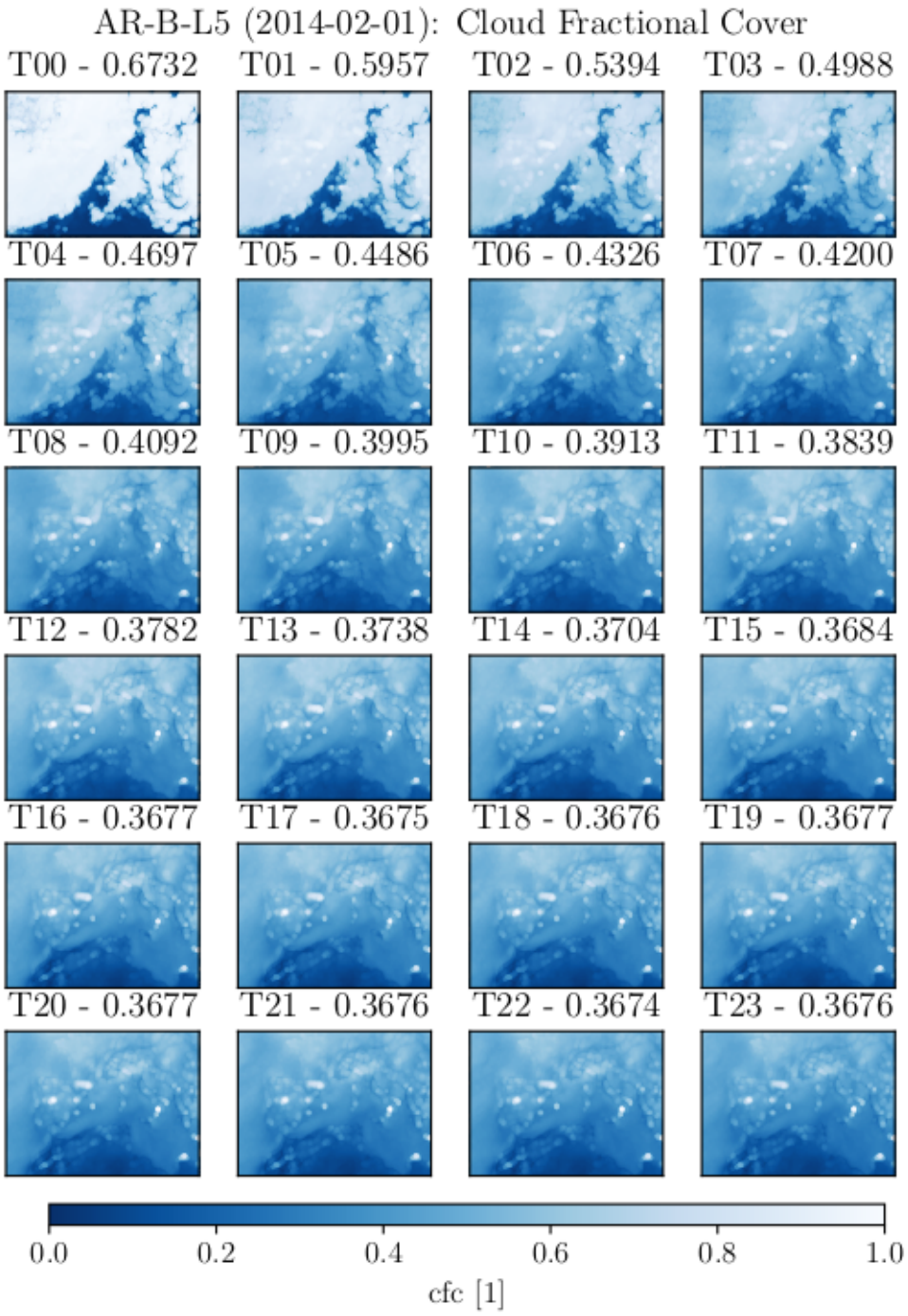


Figure F.2: *24 hour cloud cover forecast produced by $AR-B-L1$. Initiated on January 2, 2014. The area mean cloud fraction is included in the title.*

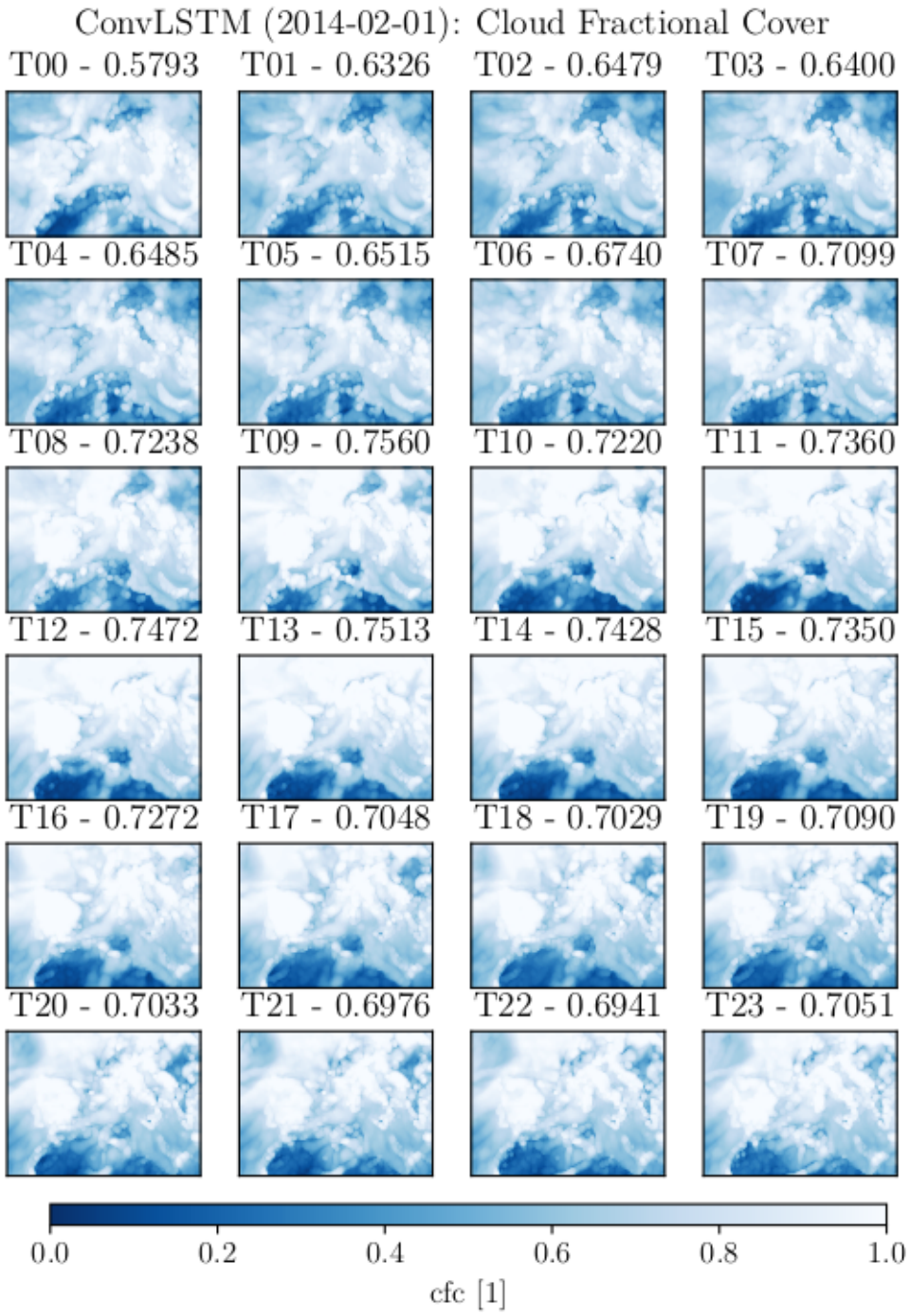


Figure F.3: *Cloud cover forecast produced by* $ConvLSTM - B_{10} - SL_{24} - 32 - 1 \times 1 - 32 - 1 \times 1$. *Initiated on January 2, 2014. The area mean cloud fraction is included in the title.*

Bibliography

Bley, S. and H. Deneke (2013). "A threshold-based cloud mask for the high-resolution visible channel of Meteosat Second Generation SEVIRI". In: *Atmospheric Measurement Techniques*, 6 (10), pp. 2713–2723. ISSN: 1867-8548. DOI: 10.5194/amt-6-2713-2013.

Bojanowski, Jdrzej S., Reto Stöckli, Anke Duguay-TeStzlaff, Stephan Finkensieper, and Rainer Hollmann (2018). "Performance assessment of the COMET Cloud Fractional Cover climatology across Meteosat generations". In: *Remote Sensing*, 10 (5). ISSN: 20724292. DOI: 10.3390/rs10050804.

Bony, Sandrine, Bjorn Stevens, Dargan Frierson, Christian Jakob, Masa Kageyama, Robert Pincus, Theodore Shepherd, Steven Sherwood, A.P. Siebesma, Adam Sobel, Masahiro Watanabe, and Mark Webb (Mar. 2015). "Clouds, circulation and climate sensitivity". In: *Nature Geoscience*, 8, pp. 261–268. DOI: 10.1038/ngeo2398.

Boucher, O., D. Randall, P. Artaxo, C. Bretherton, G. Feingold, P. Forster, V.-M. Kerminen, Y. Kondo, H. Liao, U. Lohmann, P. Rasch, S.K. Satheesh, S. Sherwood, B. Stevens, and X.Y. Zhang (2013). "Clouds and Aerosols". In: *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Ed. by T.F. Stocker, D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex, and P.M. Midgley. Cambridge, United Kingdom and New York, NY, USA: Cambridge University Press. Chap. 7, pp. 571–658. ISBN: ISBN 978-1-107-66182-0. DOI: 10.1017/CBO9781107415324.016. URL: www.climatechange2013.org.

Buitinck, Lars, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gael Varoquaux (2013). "API design for machine learning software: experiences from the scikit-learn project". In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122.

Chollet, François (2017). *Deep Learning with Python*. 1st. USA: Manning Publications Co., p. 5. ISBN: 1617294438.

– (2019). "On the Measure of Intelligence". In: pp. 1–64. URL: http://arxiv.org/abs/1911.01547.

Cox, Peter M., Chris Huntingford, and Mark S. Williamson (Jan. 2018). "Emergent constraint on equilibrium climate sensitivity from global temperature variability". In: *Nature*, 553 (7688), pp. 319–322. DOI: 10.1038/nature25450. URL: https://doi.org/10.1038/nature25450.

Danabasoglu, G., J.-F. Lamarque, J. Bacmeister, D. A. Bailey, A. K. DuVivier, J. Edwards, L. K. Emmons, J. Fasullo, R. Garcia, A. Gettelman, C. Hannay, M. M. Holland, W. G. Large, P. H. Lauritzen, D. M. Lawrence, J. T. M. Lenaerts, K. Lindsay, W. H. Lipscomb, M. J. Mills, R. Neale, K. W. Oleson, B. Otto-Bliesner, A. S. Phillips, W. Sacks, S. Tilmes, L. van Kampenhout, M. Vertenstein, A. Bertini, J. Dennis, C. Deser, C. Fischer, B. Fox-Kemper, J. E. Kay, D. Kinnison, P. J. Kushner, V. E. Larson, M. C. Long, S. Mickelson, J. K. Moore, E. Nienhouse, et al. (2020). "The Community Earth System Model Version 2 (CESM2)". In: *Journal of Advances in Modeling Earth Systems*, 12 (2). e2019MS001916 2019MS001916, e2019MS001916. DOI: 10.1029/2019MS001916. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2019MS001916. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019MS001916.

Derrien, M. and H. Le Gléau (2005). "MSG/SEVIRI cloud mask and type from SAFNWC". In: *International Journal of Remote Sensing*, 26 (21), pp. 4707–4732. DOI: 10.1080/01431160500166128. eprint: https://doi.org/10.1080/01431160500166128. URL: https://doi.org/10.1080/01431160500166128.

Drönner, Johannes, Nikolaus Korfhage, Sebastian Egli, Markus Mühling, Boris Thies, Jörg Bendix, Bernd Freisleben, and Bernhard Seeger (2018). "Fast cloud segmentation using convolutional neural networks". In: *Remote Sensing*, 10 (11), pp. 1–24. ISSN: 20724292. DOI: 10.3390/rs10111782.

Duignan, Brian (2020). *Occams razor Origin, Examples, & Facts*. en. URL: https://www.britannica.com/topic/Occams-razor (visited on 07/09/2020).

EUMETSAT (2020). *Image gallery: Mapviewer application*. https://eumetview.eumetsat.int/mapviewer/. Online; accessed 11.11.2019.

Fauske, Kjell Magne (Dec. 2006). *Example: Neural network*. http://www.texample.net/tikz/examples/neural-network/. Online; accessed 26.03.2020.

Flato, G., J. Marotzke, B. Abiodun, P. Braconnot, S.C. Chou, W. Collins, P. Cox, F. Driouech, S. Emori, V. Eyring, C. Forest, P. Gleckler, E. Guilyardi, C. Jakob, V. Kattsov, C. Reason, and M. Rummukainen (2013). "Evaluation of Climate Models". In: *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Ed. by T.F. Stocker, D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex, and P.M. Midgley. Cambridge, United Kingdom and New York, NY, USA: Cambridge University Press. Chap. 9, pp. 741–866. ISBN: ISBN 978-1-107-66182-0. DOI: 10.1017/CBO9781107415324.020. URL: www.climatechange2013.org.

Forbes, R. M. and A. M. Tompkins (2011). "An improved representation of cloud and precipitation". In: *ECMWF Newsletter*, 129 (129), pp. 13–18. DOI: 10.21957/nfgulzhe.

Forland, Gisle (Feb. 2020). *Nytt kraftig lavtrykk på vei: Kan påvirke flytrafikken i Nord-Norge*. https://www.nrk.no/nordland/nytt-lavtrykk-kan-skape-trobbel-for-wideroe-og-flytrafikken-i-nord-norge-1.14902193. Online; accessed 14.02.2020.

Fowler, Laura D. and David A. Randall (1996). *Liquid and ice cloud microphysics in the CSU general circulation model. Part III: Sensitivity to modeling assumptions*. DOI: 10.1175/1520-0442(1996)009<0561:LAICMI>2.0.CO;2.

Fujiwara, Masatomo, Jonathon S. Wright, Gloria L. Manney, Lesley J. Gray, James Anstey, Thomas Birner, Sean Davis, Edwin P. Gerber, V. Lynn Harvey, Michaela I. Hegglin, Cameron R. Homeyer, John A. Knox, Kirstin Krüger, Alyn Lambert, Craig S. Long, Patrick Martineau, Andrea Molod, Beatriz M. Monge-Sanz, Michelle L. Santee, Susann

Tegtmeier, Simon Chabrillat, David G.H. Tan, David R. Jackson, Saroja Polavarapu, Gilbert P. Compo, Rossana Dragani, Wesley Ebisuzaki, Yayoi Harada, Chiaki Kobayashi, Will McCarty, Kazutoshi Onogi, Steven Pawson, Adrian Simmons, Krzysztof Wargan, Jeffrey S. Whitaker, and Cheng Zhi Zou (2017). "Introduction to the SPARC Reanalysis Intercomparison Project (S-RIP) and overview of the reanalysis systems". In: *Atmospheric Chemistry and Physics*, 17 (2), pp. 1417–1452. ISSN: 16807324. DOI: 10.5194/acp-17-1417-2017.

Fukushima, Kunihiko (Apr. 1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". en. In: *Biological Cybernetics*, 36 (4), pp. 193–202. ISSN: 0340-1200, 1432-0770. DOI: 10.1007/BF00344251. URL: http://link.springer.com/10.1007/BF00344251 (visited on 07/09/2020).

Gers, F. A., J. Schmidhuber, and F. Cummins (1999). "Learning to forget: continual prediction with LSTM". In: *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*. Vol. 2, 850–855 vol.2.

Golaz, Jean-Christophe, Vincent E. Larson, and William R. Cotton (Dec. 2002a). "A PDF-Based Model for Boundary Layer Clouds. Part I: Method and Model Description". In: *Journal of The Atmospheric Sciences - J ATMOS SCI*, 59, pp. 3540–3551. DOI: 10.1175/1520-0469(2002)059<3540:APBMFB>2.0.CO;2.

– (2002b). "A PDF-Based Model for Boundary Layer Clouds. Part II: Model Results". In: *Journal of the Atmospheric Sciences*, 59 (24), pp. 3552–3571. DOI: 10.1175/1520-0469(2002)059<3552:APBMFB>2.0.CO;2. URL: https://doi.org/10.1175/1520-0469(2002)059%3C3552:APBMFB%3E2.0.CO;2.

Grabowski, Wojciech W., Hugh Morrison, Shin Ichiro Shima, Gustavo C. Abade, Piotr Dziekan, and Hanna Pawlowska (2019). "Modeling of cloud microphysics: Can we do better?" In: *Bulletin of the American Meteorological Society*, 100 (4), pp. 655–672. ISSN: 00030007. DOI: 10.1175/BAMS-D-18-0005.1.

Gron, Aurlien (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 1st. OReilly Media, Inc., pp. 362–365. ISBN: 1491962291.

Hersbach, H, P De Rosnay, B Bell, D Schepers, A Simmons, C Soci, S Abdalla, Alonso Balmaseda, G Balsamo, P Bechtold, P Berrisford, J Bidlot, Eric De Boisséson, M Bonavita, P Browne, R Buizza, P Dahlgren, D Dee, R Dragani, M Diamantakis, J Flemming, R Forbes, A Geer, T Haiden, E Hólm, L Haimberger, R Hogan, A Horányi, M Janisková, P Laloyaux, P Lopez, J Muñoz-Sabater, C Peubey, R Radu, D Richardson, J-N Thépaut, F Vitart, X Yang, E Zsótér, and Hao Zuo (2018). "Operational global reanalysis: progress, future directions and synergies with NWP including updates on the ERA5 production status". In: *ERA Report Series No. 27*, 27 (8th October), pp. 1–63. DOI: 10.21957/tkic6g3wm.

Hersbach, Hans, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, et al. (2020). "The ERA5 global reanalysis". In: *Quarterly Journal of the Royal Meteorological Society*. DOI: 10.1002/qj.3803. eprint: https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1002/qj.3803. URL: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.3803.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-Term Memory". In: *Neural Computation*, 9 (8), pp. 1735–1780. ISSN: 08997667. DOI: 10.1162/neco.1997.9.8.1735.

Hunter, John D. (2007). "Matplotlib: A 2D Graphics Environment". In: *Computing in Science & Engineering*, 9 (3), pp. 90–95. DOI: 10.1109/MCSE.2007.55. eprint: https://aip.scitation.org/doi/pdf/10.1109/MCSE.2007.55. URL: https://aip.scitation.org/doi/abs/10.1109/MCSE.2007.55.

Ioffe, Sergey and Christian Szegedy (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. arXiv: 1502.03167 [cs.LG].

IPCC (2013). *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge, United Kingdom and New York, NY, USA: Cambridge University Press, p. 1535. ISBN: ISBN 978-1-107-66182-0. DOI: 10.1017/CBO9781107415324. URL: www.climatechange2013.org.

Jeppesen, Jacob, Rune Jacobsen, Fadil Inceoglu, and Thomas Toftegaard (Aug. 2019). "A cloud detection algorithm for satellite imagery based on deep learning". In: *Remote Sensing of Environment*, 229, pp. 247–259. DOI: 10.1016/j.rse.2019.03.039.

Joro, Sauli, Otto Hyvärinen, and Janne Kotro (2010). "Comparison of satellite cloud masks with ceilometer sky conditions in southern Finland". In: *Journal of Applied Meteorology and Climatology*, 49 (12), pp. 2508–2526. ISSN: 15588424. DOI: 10.1175/2010JAMC2442.1.

Karlsson, K. G., E. Johansson, and A. Devasthale (2015). "Advancing the uncertainty characterisation of cloud masking in passive satellite imagery: Probabilistic formulations for NOAA AVHRR data". In: *Remote Sensing of Environment*. ISSN: 00344257. DOI: 10.1016/j.rse.2014.10.028.

Kingma, Diederik P. and Jimmy Lei Ba (2015). "Adam: A method for stochastic optimization". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15.

Kratzert, F, D Klotz, G Shalev, G Klambauer, S Hochreiter, and G Nearing (2019). "Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets". In: *Hydrology and Earth System Sciences*, 23 (12), pp. 5089–5110. DOI: 10.5194/hess-23-5089-2019. URL: https://www.hydrol-earth-syst-sci.net/23/5089/2019/.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the 25th International*

Conference on Neural Information Processing Systems - Volume 1. NIPS12. Lake Tahoe, Nevada: Curran Associates Inc., pp. 1097–1105.

Lamb, Dennis and Johannes Verlinde (Jan. 2011). *Physics and chemistry of clouds*. English (US). United Kingdom: Cambridge University Press. ISBN: 9780521899109. DOI: 10.1017/CBO9780511976377.

Lecun, Yann, Leon Bottou, Genevieve B. Orr, and Klaus-Robert Müller (1998). *Efficient BackProp*. URL: %5Curl%7Bhttp://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf%7D.

Leon, J. (May 2018). *How do I draw an LSTM cell in Tikz?* https://tex.stackexchange.com/questions/432312/how-do-i-draw-an-lstm-cell-in-tikz?rq=1. Online; accessed 26.03.2020.

Lohmann, Ulrike, Felix Lüönd, and Fabian Mahrt (2016). "Clouds". In: *An Introduction to Clouds: From the Microscale to Climate*. Cambridge University Press, pp. 19, 53–54, 81–84. DOI: 10.1017/CBO9781139087513.002.

Luo, Wenjie, Yujia Li, Raquel Urtasun, and Richard Zemel (2016). "Understanding the effective receptive field in deep convolutional neural networks". In: *Advances in Neural Information Processing Systems*, (Nips), pp. 4905–4913. ISSN: 10495258.

Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. URL: http://tensorflow.org/.

MET Norway (2020). *Pyaerocom, a Python package containing reading, post analysis and visualisation tools for the AEROCOM project*. https://github.com/metno/pyaerocom. Oslo, Norway. URL: https://pyaerocom.met.no/.

Met Office (2015). *Cartopy: a cartographic python library with a matplotlib interface*. Exeter, Devon. URL: http://scitools.org.uk/cartopy.

Nielsen, Micheal (2015). *How the backpropagation algorithm works*. Determintation press. URL: http://neuralnetworksanddeeplearning.com/chap2.html.

Olah, Christopher (Aug. 2015). *Understanding LSTM Networks*. http://colah.github.io/posts/2015-08-Understanding-LSTMs/. Online; accessed 26.03.2020.

Parker, Wendy S. (2016). "Reanalyses and observations: What's the Difference?" In: *Bulletin of the American Meteorological Society*, 97 (9), pp. 1565–1572. ISSN: 00030007. DOI: 10.1175/BAMS-D-14-00226.1.

Ramanathan, V., R. D. Cess, E. F. Harrison, P. Minnis, B. R. Barkstrom, E. Ahmad, and D. Hartmann (Jan. 1989). "Cloud-Radiative Forcing and Climate: Results from the Earth Radiation Budget Experiment". In: *Science*, 243 (4887), pp. 57–63. DOI: 10.1126/science.243.4887.57. URL: https://doi.org/10.1126/science.243.4887.57.

Rodrigues, Eduardo Rocha, Igor Oliveira, Renato Cunha, and Marco Netto (2018). "DeepDownscale: A deep learning strategy for high-resolution weather forecast". In: *Proceedings - IEEE 14th International Conference on eScience, e-Science 2018*, pp. 415–422. DOI: 10.1109/eScience.2018.00130.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). "Learning Internal Representations by Error Propagation". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, pp. 318–362. ISBN: 026268053X.

Schmetz, Johannes, Paolo Pili, Stephen Tjemkes, Dieter Just, Jochen Kerkmann, Sergio Rota, and Alain Ratier (2002). "AN INTRODUCTION TO METEOSAT SECOND GENERATION (MSG)". In: *Bulletin of the American Meteorological Society*, 83 (7), pp. 977–

992. DOI: 10.1175/1520-0477(2002)083<0977:AITMSG>2.3.CO;2. URL: https://doi.org/10.1175/1520-0477(2002)083%3C0977:AITMSG%3E2.3.CO;2.

Schulzweida, Uwe (Oct. 2019). *CDO User Guide*. DOI: 10.5281/zenodo.3539275. URL: https://doi.org/10.5281/zenodo.3539275.

Seland, Ø., M. Bentsen, L. Seland Graff, D. Olivié, T. Toniazzo, A. Gjermundsen, J. B. Debernard, A. K. Gupta, Y. He, A. Kirkevåg, J. Schwinger, J. Tjiputra, K. Schancke Aas, I. Bethke, Y. Fan, J. Griesfeller, A. Grini, C. Guo, M. Ilicak, I. H. Hafsahl Karset, O. Landgren, J. Liakka, K. Onsum Moseid, A. Nummelin, C. Spensberger, H. Tang, Z. Zhang, C. Heinze, T. Iverson, and M. Schulz (2020). "The Norwegian Earth System Model, NorESM2 – Evaluation of theCMIP6 DECK and historical simulations". In: *Geoscientific Model Development Discussions*, 2020, pp. 1–68. DOI: 10.5194/gmd-2019-378. URL: https://gmd.copernicus.org/preprints/gmd-2019-378/.

Shi, Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo (2015). *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting*. arXiv: 1506.04214 [cs.CV].

Simmons, Adrian, Jean Louis Fellous, Venkatachalam Ramaswamy, Kevin Trenberth, Ghassem Asrar, Magdalena Balmaseda, John P. Burrows, Philippe Ciais, Mark Drinkwater, Pierre Friedlingstein, Nadine Gobron, Eric Guilyardi, David Halpern, Martin Heimann, Johnny Johannessen, Pieternel F. Levelt, Ernesto Lopez-Baeza, Joyce Penner, Robert Scholes, and Ted Shepherd (2016). "Observation and integrated Earth-system science: A roadmap for 2016-2025". In: *Advances in Space Research*, 57 (10), pp. 2037–2103. ISSN: 18791948. DOI: 10.1016/j.asr.2016.03.008.

stackexchange (Aug. 2019a). *Architecture Neural Network with weights*. https://tex.stackexchange.com/questions/505741/architecture-neural-network-with-weights. Online; accessed 26.03.2020.

– (Dec. 2019b). *Visualizing matrix convolution*. https://tex.stackexchange.com/questions/522118/visualizing-matrix-convolution. Online; accessed 26.03.2020.

Stephens, Graeme, David Winker, Jacques Pelon, Charles Trepte, Deborah Vane, Cheryl Yuhas, Tristan L'Ecuyer, and Matthew Lebsock (2018). "Cloudsat and calipso within the a-train: Ten years of actively observing the earth system". In: *Bulletin of the American Meteorological Society*, 99 (3), pp. 569–581. ISSN: 00030007. DOI: 10.1175/BAMS-D-16-0324.1.

Stöckli, Reto, Jdrzej S. Bojanowski, Viju O. John, Anke Duguay-Tetzlaff, Quentin Bourgeois, Jörg Schulz, and Rainer Hollmann (2019). "Cloud detection with historical geostationary satellite sensors for climate applications". In: *Remote Sensing*, 11 (9), pp. 1–21. ISSN: 20724292. DOI: 10.3390/rs11091052.

Stubenrauch, C. J., W. B. Rossow, S. Kinne, S. Ackerman, G. Cesana, H. Chepfer, L. Di Girolamo, B. Getzewich, A. Guignard, A. Heidinger, B. C. Maddux, W. P. Menzel, P. Minnis, C. Pearl, S. Platnick, C. Poulsen, J. Riedi, S. Sun-Mack, A. Walther, D. Winker, S. Zeng, and G. Zhao (2013). "Assessment of global cloud datasets from satellites: Project and database initiated by the GEWEX radiation panel". In: *Bulletin of the American Meteorological Society*, 94 (7), pp. 1031–1049. ISSN: 00030007. DOI: 10.1175/BAMS-D-12-00117.1.

Sun, Shuo and Gaoxiang Liu (2018). *Air Quality Forecasting Using Convolutional LSTM*. URL: http://cs230.stanford.edu/projects_spring_2018/reports/8291197.pdf.

Taravat, Alireza, Simon Proud, Simone Peronaci, Fabio Del Frate, and Natascha Oppelt (2015). "Multilayer perceptron neural networks model for meteosat second generation SEVIRI daytime cloud masking". In: *Remote Sensing*, 7 (2), pp. 1529–1539. ISSN: 20724292. DOI: 10.3390/rs70201529.

Tiedtke, M. (Nov. 1993). "Representation of Clouds in Large-Scale Models". In: *Monthly Weather Review*, 121 (11), pp. 3040–3061. ISSN: 0027-0644. DOI: 10.1175/1520-0493(1993)121<3040:ROCILS>2.0.CO;2. eprint: https://journals.ametsoc.org/mwr/article-pdf/121/11/3040/4171932/1520-0493(1993)121_3040_rocils_2_0_co_2.pdf. URL: https://doi.org/10.1175/1520-0493(1993)121%3C3040:ROCILS%3E2.0.CO;2.

Tomkins, Adrian M (2005). *The parametrization of cloud cover*. https://www.ecmwf.int/sites/default/files/elibrary/2005/16958-parametrization-cloud-cover.pdf?fbclid=IwAR3BlzrOuNF0v3bYcxSaMT19u5PhcD8DDTRzOpp-BspnSPwXhZOBn3wUj48.

Tompkins, Adrian M (2009). "Cloud parametrization". In: *Seminar on Parametrization of Subgrid Physical Processes, 1–4 September 2008*, (September), pp. 27–62. URL: http://www.ecmwf.int/publications/library/do/references/show?id=88790.

Voosen, Paul (2018). "Science insurgents plot a climate model driven by artificial intelligence". In: *Science*. ISSN: 0036-8075. DOI: 10.1126/science.aau8974.

Wang, Xuanji and Jeffrey R. Key (July 2005). "Arctic Surface, Cloud, and Radiation Properties Based on the AVHRR Polar Pathfinder Dataset. Part I: Spatial and Temporal Characteristics". In: *Journal of Climate*, 18 (14), pp. 2558–2574. ISSN: 0894-8755. DOI: 10.1175/JCLI3438.1. eprint: https://journals.ametsoc.org/jcli/article-pdf/18/14/2558/3787635/jcli3438_1.pdf. URL: https://doi.org/10.1175/JCLI3438.1.

Waskom, Michael, Olga Botvinnik, Paul Hobson, John B. Cole, Yaroslav Halchenko, Stephan Hoyer, Alistair Miles, Tom Augspurger, Tal Yarkoni, Tobias Megies, Luis Pedro Coelho, Daniel Wehner, cynddl, Erik Ziegler, diego0020, Yury V. Zaytsev, Travis Hoppe, Skipper Seabold, Phillip Cloud, Miikka Koskinen, Kyle Meyer, Adel Qalieh, and Dan Allan (Nov. 2014). *seaborn: v0.5.0 (November 2014)*. Version v0.5.0. DOI: 10.5281/zenodo.12710. URL: https://doi.org/10.5281/zenodo.12710.

Wild, Martin, Maria Z. Hakuba, Doris Folini, Patricia Dörig-Ott, Christoph Schär, Seiji Kato, and Charles N. Long (2019). "The cloud-free global energy balance and inferred cloud radiative effects: an assessment based on direct observations and climate models". In: *Climate Dynamics*, 52 (7-8), pp. 4787–4812. ISSN: 14320894. DOI: 10.1007/s00382-018-4413-y. URL: http://dx.doi.org/10.1007/s00382-018-4413-y.

Zhong, Wenyi and Joanna D. Haigh (2013). "The greenhouse effect and carbon dioxide". In: *Weather*, 68 (4), pp. 100–105. DOI: 10.1002/wea.2072. eprint: https://rmets.

onlinelibrary.wiley.com/doi/pdf/10.1002/wea.2072. URL: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/wea.2072.

Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros (2017). *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. arXiv: 1703.10593 [cs.CV].