Improved Weather Forecasting using AI

Project Report

Submitted by

Mr. Vaisakh S B
20-27-01

Under the kind guidance of

Dr. Somanchi. V. S. S. N. V. G. Krishna Murthy (Associate Professor and HoD, Department of Applied Mathematics, DIAT, Pune)
Dr. Bipin Kumar (Scientist 'E', IITM Pune)

Department of Applied Mathematics
Defence Institute of Advanced Technology
Girinagar, Next to Khadakwasla Dam, Pune,Maharashtra-411025
(Affiliated to Ministry of Defence)
Dec 2021

Contents

List of Figures

List of Tables

Abstract

In the field of meteorology and climate science, it is important to understand how different climate variables interact. Also identifying the relationship between one variable with another variable can benefit climate science and humankind. Here, in this study, we are using a deep learning CNN architecture called UNET to understand and learn the pattern between input multi-variables and the target variable. The results are then compared with the operational dynamical model and the target variables are forecasted. In this study the target variables being Leaf Area Index, Fraction of Absorbed Photosynthetically Active Radiation, Fire Weather Index, Surface Air Temperature, Mass concentration of chlorophyll A, Mixed Layer Depth, Precipitation, Sea Water Salinity, Sea Surface Temperature, Sea Surface Height. This shows that the UNET architecture can untie the physical relationships to the target variables, which can be used in the operational dynamical model which in turn improves the forecasts with greater accuracy

Chapter 1

Introduction

Climate science and meteorology have been gaining enormous importance in recent times. It affects all living beings in many ways. Therefore it is important to analyze the weather conditions and forecasting with greater accuracy plays an important role in preventing extreme conditions and prevents a disaster. The physics-based data-driven models that we now have are prone to large error values. We can't totally nullify this error range, so this should be narrowed down to get better forecasts.

Artificial Intelligence (AI), the capability of a machine to perform tasks to accomplish a specific objective based on the provided data, is revolutionizing pretty much every field known to humankind. Machine Learning, a subset of AI, has advanced so much that it is being used even in anomaly detection, which was not possible 15-20 years ago. Deep Learning, termed as a data-hungry method as it needs a larger dataset, can understand and learn the mapping of input variables and target variables. This can be used in the area of forecasting as well. The advancements of open-source libraries such as TensorFlow, Keras, etc with high-performance computing had lowered the barriers of complex computations. With high-performance computing and parallel processing, we can now load larger datasets than before.

This report focuses on one such deep learning model UNET based on CNN to understand and learn the mapping between the input variables and the target climatic variables and forecasts the target variables into the future. At this point in time, only three problem statements are well defined. They are forecasting the variables Precipitation, Sea Surface Temperature and Sea Surface Height. The estimation of precipitation in the state of art weather and climate models is an important process. "The real holes in climate science" [2] had categorized this as one of the four major problems of advancement in weather and climate science. Precipitation along with other climate variables contributes to the formation of rainfall. Therefore, we can forecast the rainfall across the globe.

Sea Surface Temperature (SST) is an important parameter in the energy balance system of the earth's surface and it is also a critical indicator to measure the heat of the seawater. The phenomenon of ENSO (El Niño–Southern Oscillation) and several other processes depend on the SST. Sea Surface Height (SSH) prediction is considered theoretically and practically significant for regional and global ocean-related research. The area of oceanic variables forecasting has traditionally relied on models that are numerical models. Hence with deep learning, there is a chance of higher accuracy than the traditional methods. We can even find out which all cities will be underwater in the distant future and plan cities accordingly to evade a disaster.

Apart from these, there are 7 more target variables named Leaf Area Index, Fraction of Absorbed Photosynthetically Active Radiation, Fire Weather Index, Surface Air Temperature, Mass concentration of chlorophyll A, Mixed Layer Depth, Sea Water Salinity. More problem statements will be added as the project continues.

Chapter 2

Literature Survey

2.1 Univariate models

Chattopadhyay et al. [8] predicted the monthly maximum air temperature on northeastern India using three different ANN methodologies named MLP (multi layer perceptron), MNN (modular neural networks) and GFFNN (generalized feed forward neural network). The data used was from 1901 to 2003. Air temperature maximum values of previous months were used as inputs to the model network. In their study, MNN model outperforms the MLP and GFFNN model.

Kumar et al. [5] used a neural network architecture called FFNN (feed forward neural network) to forecast the mean air temperature on weekly basis in India. Previous six weeks air temperature data is used as an input to the architecture and forecast air temperature with lead time as 1-week. In this study they tuned the architecture model configuration on the basis of assessing RMSE and R^2 metrices. As a result, a two-hidden-layer model with 5 neurons on each layer was found to give best results.

Other studies focussed on deep learning architectures rather than traditional neural network architectures. Zhang et al. [9] forecasted daily air temperature average for a lead time of 4 days ahead by using a deep learning architecture called CRNN (convolutional recurrent neural networks), which is a combination of CNNs (convolutional neural networks) and RNNs (recurrent neural networks). They provided the data from 1952 to 2018 daily air temperature data on China as to train the model. The study resulted that their model is able to successfully predict air temperature based on the previous air temperature data.

Li et al. [10] used a complex deep learning architecture called Stacked LSTM (long short term memory) network which is a LSTM model comprising multiple LSTM layers. This study predicted half-hourly lead time air temperature with its historical observations as inputs to the

defined model. The proposed model had 3 hidden layers with 20, 10 and 4 memory cells in each of the layers and fully connected and output layer with 4 and 1 neurons, respectively. They compared the model with baselines as DNN (deep neural network) architecture and RF (random forest) under different sliding windows. The result claimed that their model is superior than their baseline models.

In [6], deep convolutional neural network (DLWP) known as UNET is applied to 500-hpa geopotential height reanalysis dataset. Further improved version [7] (DLWP-CS) transformed the spherical global data to the cubed sphere and showed improved predictions. This helped to reduce and minimize the error caused by the spherical distortion of the data. The [1] discussed the use of modified DLWP-CS UNET architecture to predict the global precipitation forecast with good results. The study used data from 1980 to 2015 with total cloud cover, precipitation and solar radiation as the predictors and precipitation being the predictand.

2.2 Multivariate models

Smith et al. [11] employed a Ward-style ANN model to forecast air temperature with a lead time of 1-12 h ahead. They incorporated air temperature hourly data, wind stress, humidity, solar radiation and precipitation as their inputs to the model. The data from 1997-2005 were used in the process. They tried with ensemble methods also but didn't found out to be good than their base model.

Akram and El [12] used a deep LSTM network to predict air temperature, humidity and wind stress with lead time of 24 or 72 hrs in 9 cities in Morocco using the 24 or 72 hrs previous hourly data of air temperature, humidity and wind stress as predictors or the input to the model. Their study had a fully connected hidden layer with 100 neurons between two LSTM layers. They were able to forecast air temperature with greated accuracy.

Sundaram et al. [13] compared three different machine learning models namely, MLP, RNN and SVM (support vector machine) for air temperature daily forecasting. Different predictors were used as an input to the model named air temperature, wind, atmospheric pressure, dew point temperature, relative humidity and total cloud cover. The meteorological data was from 2006 till 2018.The study found that RNN was the best among the three models.

Kreuzer et al. [14] applied convLSTM (convolutional long short-term memory) method to forecast air temperature with a lead time of 24h in advance for several weather stations in Germany. The data used was from 2009 till 2018. This study compared the performance of convLSTM with the SARIMA (seasonal autoregressive integrated moving average), univariate and multivariate LSTMs. Hourly air temperature, relative humidity, cloud coverage, precipitation, wind speed

and direction, month of year, hour of day, sealevel air pressure, and the difference between the air pressure at the station and the sea level were used as inputs in multivariate LSTM and ConvLSTM. They showed that the seasonal naive approach has the worst performance for most of the prediction horizons. While the SARIMA and univariate LSTM network performed well for the first two- to three-hour air forecasts, the ConvLSTM and multivariate LSTM showed a better performance for longer forecast horizons. In the stations with large variations of air temperature during the day, convLSTM outperformed other methods.

Lee et al. [15] used three different named MLP, LSTM and CNN to forecast maximum, minimum and average air temperature with lead time as 3 days in regions of South Korea. They tried both hourly and daily air temperature, cloud cover, ground surface temperature, humidity, vapor pressure, dew point temperature, atmospheric pressure, sea-level pressure, precipitation, hours of sunshine, solar radiation, and wind speed and direction as inputs in the previous 30 days. This study showed that CNN outperforms MLP and LSTM. Also it is found that hourly input data provided better information on daily air temperature forecasting than with daily input data.

Study	**Data Span**	**Model**	**A**	**B**	**C**	**D**	**E**
[6]	1979-2010	DLWP	✓	✗	✗	✓	✓
[7]	1979-2018	DLWP-CS	✓	✓	✗	✓	✓
[1]	1980-2015	Modified DLWP-CS	✓	✓	✗	✗	✓
[8]	1901-2003	MLP, GFFNN and MNN	✗	✗	✓	✓	✗
[5]	2002-2011	FFANN	✗	✗	✓	✓	✗
[9]	1952-2018	CRNN	✗	✗	✓	✓	✗
[10]	2009-2018	Stacked LSTM	✗	✗	✓	✓	✗
[11]	1997-2005	Ward-Style ANN	✗	✗	✓	✗	✗
[12]	2000-2015	LSTM	✗	✗	✓	✗	✗
[13]	2006-2018	MLP	✗	✗	✓	✗	✗
[14]	2009-2018	Multivariate LSTM, ConvLSTM	✗	✗	✓	✗	✗
[15]	2009-2018	MLP, LSTM, CNN	✗	✗	✓	✗	✗
This Project	1961-2020	Modified DLWP-CS	✓	✓	✓	✓	✓

Abbreviations: ***A:*** Global datasets ***B:*** Remove/minimizes spherical distortion error ***C:*** Surface air temperature as predictand ***D:*** Univariate ***E:*** Comparison with real operational system

Table 2.1: Description of selected studies

Chapter 3

Meteorological Data Format and Tools

For climate model generated data the most commonly used data format is the netCDF format (".nc"). NetCDF (Network Common Data Form) facilitates access to array oriented scientific data. [Figure 3.1] is a sample 'dump' of a typical netCDF file.

```
netcdf remap_air_2000_2009 {
dimensions:
        time = UNLIMITED ; // (3653 currently)
        lon = 144 ;
        lat = 143 ;
variables:
        double time(time) ;
                time:standard_name = "time" ;
                time:long_name = "Time" ;
                time:units = "days since 2000-01-01 00:00:00" ;
                time:calendar = "proleptic_gregorian" ;
                time:axis = "T" ;
        float lon(lon) ;
                lon:standard_name = "longitude" ;
                lon:long_name = "Longitude" ;
                lon:units = "degrees_east" ;
                lon:axis = "X" ;
        float lat(lat) ;
                lat:standard_name = "latitude" ;
                lat:long_name = "Latitude" ;
                lat:units = "degrees_north" ;
                lat:axis = "Y" ;
        float air(time, lat, lon) ;
                air:long_name = "mean Daily Air temperature at 2 m" ;
                air:units = "degK" ;
                air:_FillValue = NaNf ;
                air:missing_value = NaNf ;
                air:least_significant_digit = 1s ;
                air:precision = 2s ;
                air:GRIB_id = 11s ;
                air:GRIB_name = "TMP" ;
                air:var_desc = "Air temperature" ;
                air:dataset = "NCEP Reanalysis Daily Averages" ;
                air:level_desc = "2 m" ;
                air:statistic = "Mean" ;
                air:parent_stat = "Individual Obs" ;
                air:actual_range = 178.5f, 316.07f ;

// global attributes:
                :CDI = "Climate Data Interface version 1.9.10 (https://mpimet.mpg.de/cdi)" ;
                :Conventions = "COARDS" ;
                :title = "mean daily NMC reanalysis (2000)" ;
                :description = "Data is from NMC initialized reanalysis\n(4x/day).  It consists of T62 variables interpolated to\npressure surfaces from model (sigma)
 surfaces." ;
                :platform = "Model" ;
                :history = "Sun Aug 15 00:24:55 2021: cdo remapbil,/lus/dal/mtechstudent/MLD/Vaisakh/cmip6_data/tas/nc/tas_day_IPSL-CM6A-LR_dcppA-hindcast_s1999-r1i1p
1f1_gr_20000101-20091231.nc /lus/dal/mtechstudent/MLD/Vaisakh/cmip6_data/air/nc/air_2000_2009.nc remap_air_2000_2009.nc\ncreated 00/01/30 by Hoop (netCDF2.3)\nConvert
ed to chunked, deflated non-packed NetCDF4 2014/09" ;
                :dataset_title = "NCEP-NCAR Reanalysis 1" ;
                :References = "http://www.psl.noaa.gov/data/gridded/data.ncep.reanalysis.html" ;
                :CDO = "Climate Data Operators version 1.9.10 (https://mpimet.mpg.de/cdo)" ;
}
```

Figure 3.1: dump result

It has several components. Dimension names, dimension sizes, Variables available, fill values, missing values details, coordinate variables, etc. In the following, time(time), lat(lat) and lon(lon)

are classified as coordinate variables while air(time, lat, lon) is classified as variables. Data in netCDF format is:

- Self-describing: It includes meta data information.
- Scalable: Any part of the data is easily accessable using suitable engines and also even by remote access also.
- Easily Appendable: Data can be easily appended to the existing dataset without modifing or redefining it.

For manipulating and processing these files a python package and open source project named xarray can be used. This makes working with labelled multi-dimensional arrays efficient and simple. Xarray is built on top of NumPy. Thus it is more intuitive and concise. This also includes visualization tools. Xarray also borrows heavily from pandas, which is helpful in tabular data. Finally, xarray integrates tightly with dask for parallel computing, making it efficient in all the way possible.

```
>>> import xarray as xr
>>> ds = xr.open_dataset("remap_air_2000_2009.nc")
>>> ds
<xarray.Dataset>
Dimensions:  (lat: 143, lon: 144, time: 3653)
Coordinates:
  * time     (time) datetime64[ns] 2000-01-01 2000-01-02 ... 2009-12-31
  * lon      (lon) float32 0.0 2.5 5.0 7.5 10.0 ... 350.0 352.5 355.0 357.5
  * lat      (lat) float32 -90.0 -88.73 -87.46 -86.2 ... 86.2 87.46 88.73 90.0
Data variables:
    air      (time, lat, lon) float32 ...
Attributes:
    CDI:            Climate Data Interface version 1.9.10 (https://mpimet.mpg...
    Conventions:    COARDS
    title:          mean daily NMC reanalysis (2000)
    description:    Data is from NMC initialized reanalysis\n(4x/day).  It co...
    platform:       Model
    history:        Sun Aug 15 00:24:55 2021: cdo remapbil,/lus/dal/mtechstud...
    dataset_title:  NCEP-NCAR Reanalysis 1
    References:     http://www.psl.noaa.gov/data/gridded/data.ncep.reanalysis...
    CDO:            Climate Data Operators version 1.9.10 (https://mpimet.mpg...
```

Figure 3.2: data read by xarray

[Figure 3.2] shows how to read xarray to read netcdf files and display its variables, coordinate variables and attributes. The method xr.open_dataset() allows you to read a single netCDF file

and xr.open_mfdataset() allows user to read multiple netCDF files all at once and even supports dask parallel computing so that we can load data as chunks rather than loading as a whole. It is easy to manipulate netCDF file using xarray. [Figure 3.3] helps to look the coordinate variables and attributes using ds.coords and ds.attrs methods.

```
>>> ds.co
ds.coarsen(        ds.combine_first(  ds.compute(        ds.conj(           ds.conjugate(      ds.coords          ds.copy(           ds.count(
>>> ds.co
ds.coarsen(        ds.combine_first(  ds.compute(        ds.conj(           ds.conjugate(      ds.coords          ds.copy(           ds.count(
>>> ds.coords
Coordinates:
  * time     (time) datetime64[ns] 2000-01-01 2000-01-02 ... 2009-12-31
  * lon      (lon) float32 0.0 2.5 5.0 7.5 10.0 ... 350.0 352.5 355.0 357.5
  * lat      (lat) float32 -90.0 -88.73 -87.46 -86.2 ... 86.2 87.46 88.73 90.0
>>> ds.attrs
{'CDI': 'Climate Data Interface version 1.9.10 (https://mpimet.mpg.de/cdi)', 'Conventions': 'COARDS', 'title': 'mean daily NMC reanalysis (2000)', 'description': 'Dat
a is from NMC initialized reanalysis\n(4x/day).  It consists of T62 variables interpolated to\npressure surfaces from model (sigma) surfaces.', 'platform': 'Model', '
history': 'Sun Aug 15 00:24:55 2021: cdo remapbil,/lus/dal/mtechstudent/MLD/Vaisakh/cmip6_data/tas/nc/tas_day_IPSL-CM6A-LR_dcppA-hindcast_s1999-r1i1p1f1_gr_20000101-2
0091231.nc /lus/dal/mtechstudent/MLD/Vaisakh/cmip6_data/air/nc/air_2000_2009.nc remap_air_2000_2009.nc\ncreated 00/01/30 by Hoop (netCDF2.3)\nConverted to chunked, de
flated non-packed NetCDF4 2014/09', 'dataset_title': 'NCEP-NCAR Reanalysis 1', 'References': 'http://www.psl.noaa.gov/data/gridded/data.ncep.reanalysis.html', 'CDO':
'Climate Data Operators version 1.9.10 (https://mpimet.mpg.de/cdo)'}
```

Figure 3.3: coordinate variables

Chapter 4

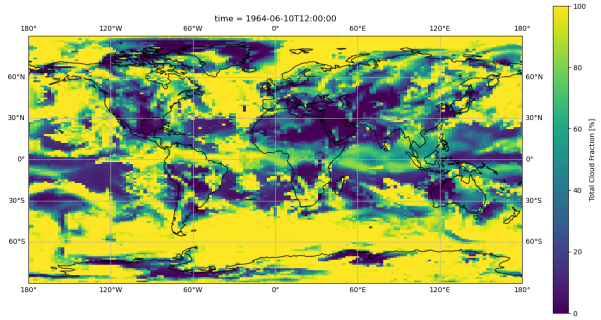
Data Preprocessing

4.1 Model variables OR Predictors

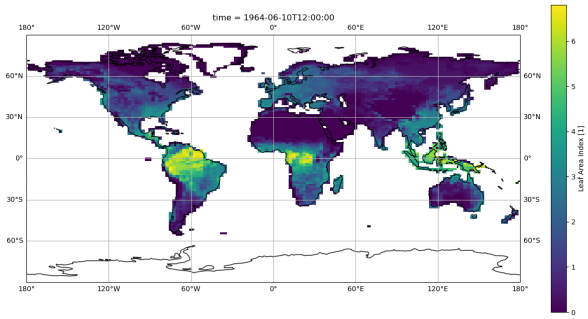
	Variable	Standard Name	Long Name	Units	Resolution
0	clt	cloud_area_fraction	Total Cloud Fraction	%	(143, 144)
1	hfls	surface_upward_latent_heat_flux	Surface Upward Latent Heat Flux	W m-2	(143, 144)
2	hfss	surface_upward_sensible_heat_flux	Surface Upward Sensible Heat Flux	W m-2	(143, 144)
3	lai	leaf_area_index	Leaf Area Index	1	(143, 144)
4	mrsos	mass_content_of_water_in_soil_layer	Moisture in Upper Portion of Soil Column	kg m-2	(143, 144)
5	pr	precipitation_flux	Precipitation	kg m-2 s-1	(143, 144)
6	qnet	NaN	NaN	NaN	(143, 144)
7	sos	sea_surface_salinity	Sea Surface Salinity	0.001	(180, 360)
8	t20d	depth_of_isosurface_of_sea_water_potential_tem...	20C isotherm depth	m	(180, 360)
9	ta700	air_temperature	Air Temperature	K	(143, 144)
10	ta850	air_temperature	Air Temperature	K	(143, 144)
11	tas	air_temperature	Near-Surface Air Temperature	K	(143, 144)
12	tos	sea_surface_temperature	Sea Surface Temperature	degC	(180, 360)
13	uas	eastward_wind	Eastward Near-Surface Wind	m s-1	(143, 144)
14	vas	northward_wind	Northward Near-Surface Wind	m s-1	(143, 144)
15	wap500	lagrangian_tendency_of_air_pressure	omega (=dp/dt)	Pa s-1	(143, 144)

Table 4.1: Model variables

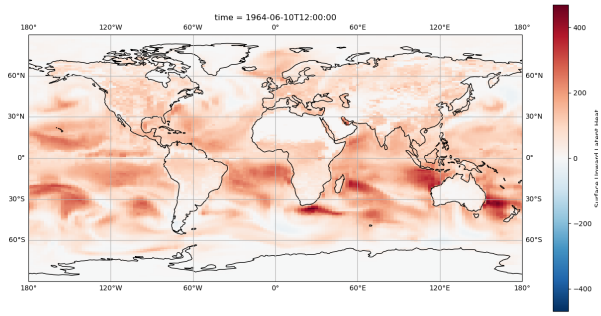
We have a total of 16 input model variables [Table 1]. All the model variables are hindcast model data. Hindcast is a numerical weather prediction model (physics-based model running on partial differential equations) which is started from the exact observational date. The outputs from these models are presently used as forecasts, which goes to the public. Hindcast has been running for



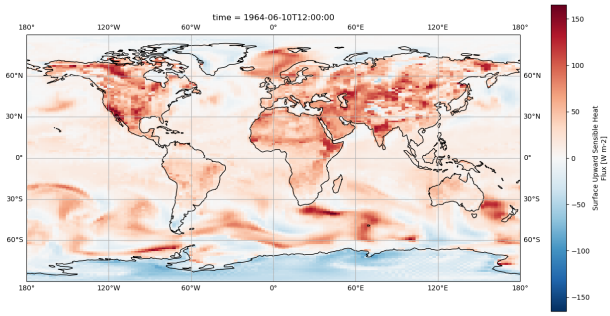
clt



lai



hfls

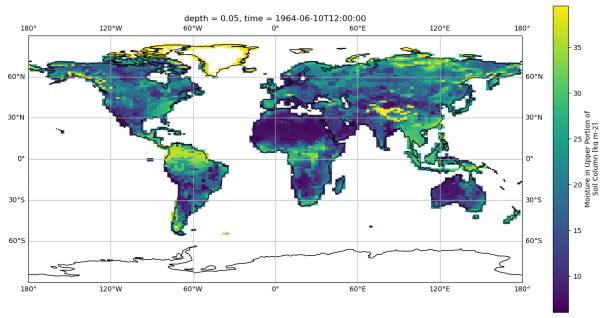


hfss

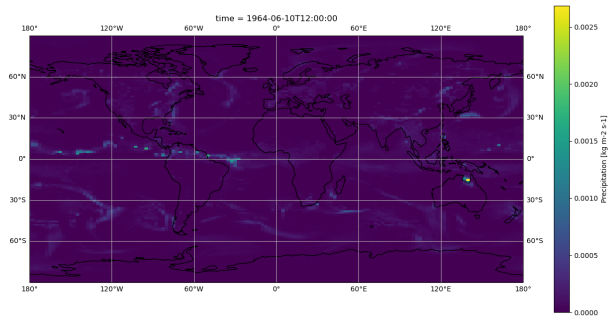
the past many years. Also, the data is daily CMIP6 (the Sixth Phase of the Coupled Model Intercomparison Project) data.

From the Taylor diagram of s - p patterns in [4], models CMCC-CMS, MPI-ESM-P performed well with CMIP5 data. But at the time of data gathering on 23rd June 2021, it has found that not all the input variables were available for the above models. Variables sos and tos have been downloaded from source CanESM5 and all the remaining input variables from the source IPSL-CM6A-LR. The data for the variable qnet (Net surface heat flux) is not available readily for CMIP6 data. However, it can be derived from rlds (surface downwelling longwave flux in air), rlus (surface upwelling longwave flux in air), rsds (surface downwelling shortwave flux in air) and rsus (surface upwelling shortwave flux in air).

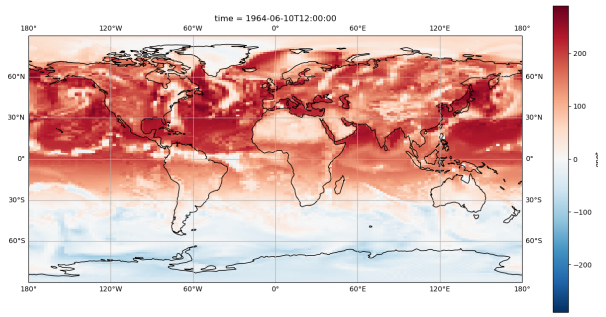
$$qnet = rsds - rsus + rlds - rlus$$



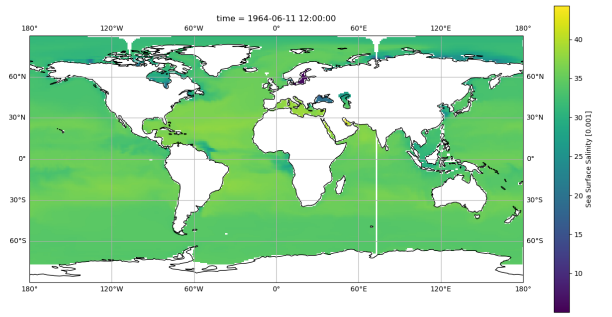
mrsos



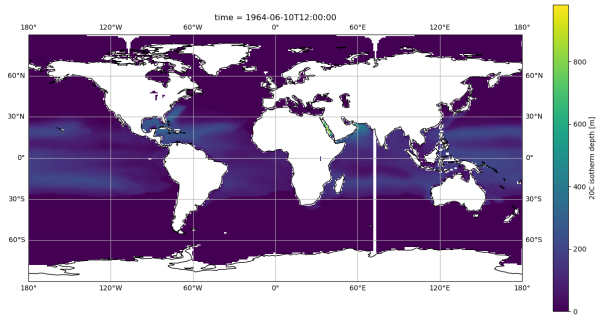
pr



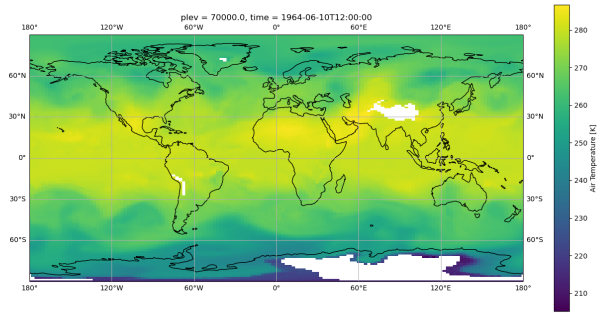
qnet



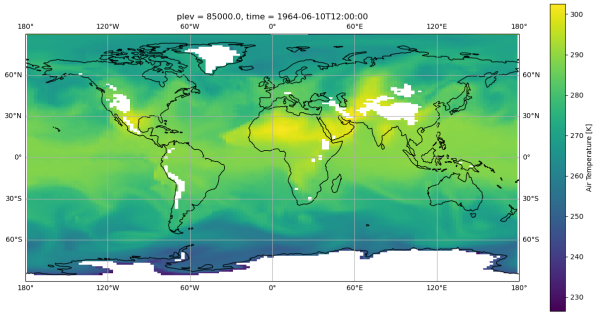
sos



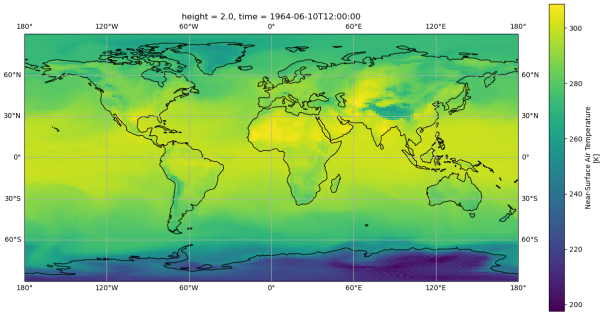
t20d



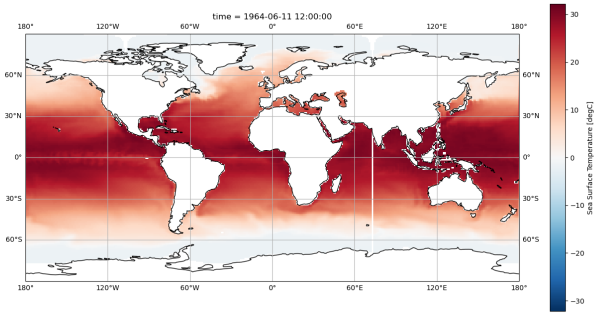
ta700



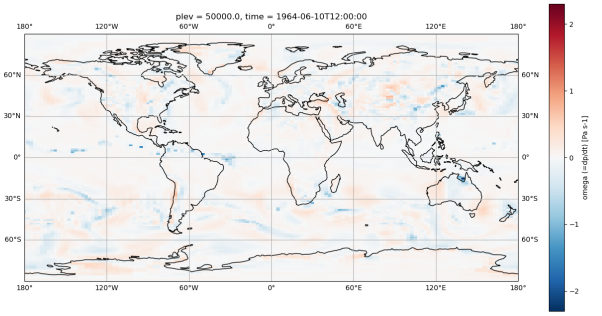
ta850



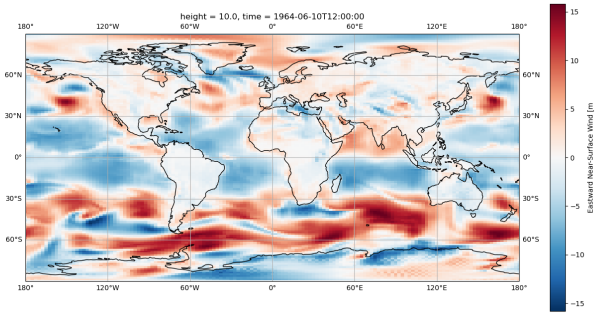
tas



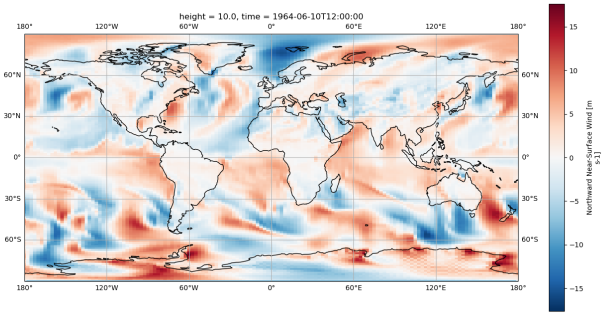
tos



wap500



uas



vas

Figure 4.3: Model Variables

4.2 Preprocessing predictors

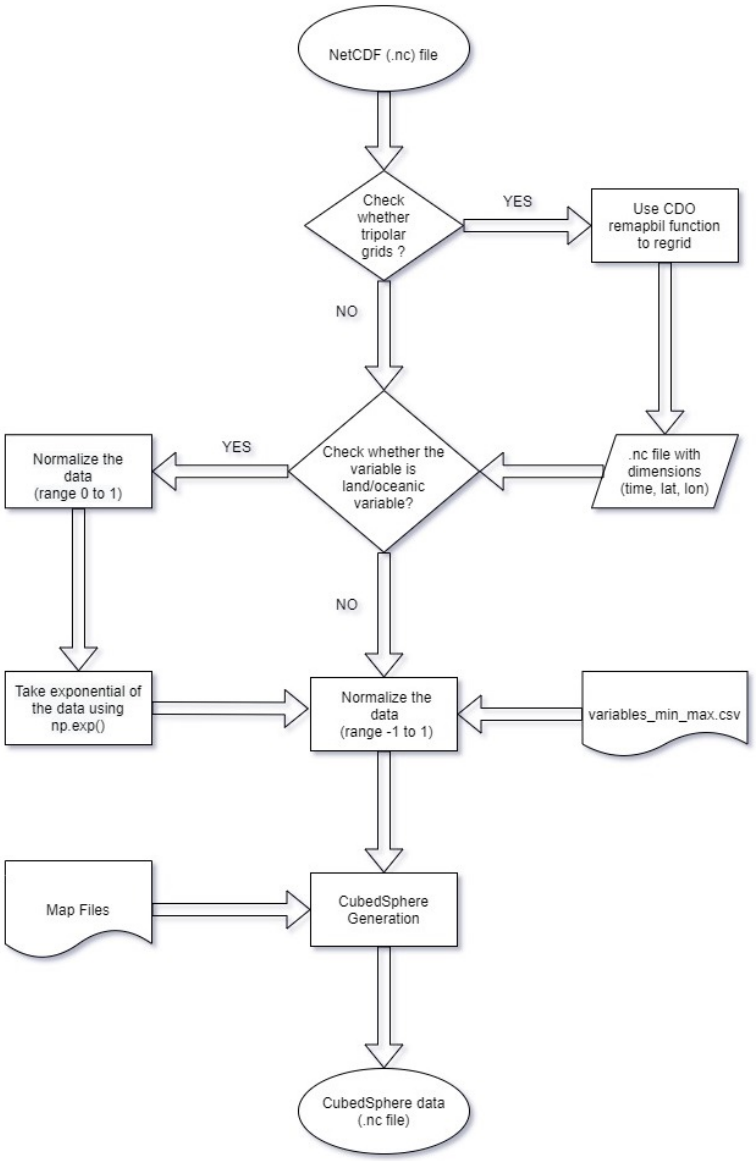


Figure 4.4: flowchart of preprocessing data

The CMIP6 hindcast data is downloaded as a batch of 10 years files. The hindcast model will take a year and it will run for 10 years and the data is stored. For every input variable, we have data from 1961 to 2016. Since it is in batches of 10, we have till 2026 into the future. A total of 40 members have uploaded the CMIP6 data. We have downloaded the data with varient_label r1i1p1f1.

Variables sos, tos and t20d were found to be in a tripolar grid rather than the normal lat lon

format. Since all our input variables and target variables are in general lat lon coordinate format, there is a need of changing the tripolar grid to lat lon format. Climate Data Operator (CDO) can accomplish this task using the bilinear remapping function.

Land	Ocean	Atmospheric
lai	sos	clt
mrsos	t20d	hfls
	tos	hfss
		pr
		qnet
		ta700
		ta850
		tas
		uas
		vas
		wap500

Table 4.2: Realm of predictors

Further processing depends on the realm of the variable. We need to mask the land area. Therefore for masking the land and oceanic variables we are taking the exponential of the data array. Since we are taking exponential for the land and oceanic variables, the resulting data array's value range becomes so huge, can be even to the power of 100s. So in order to reduce this to a smaller value range, we use min-max normalization on the data initially. This will reduce the values into the range 0-1. Each variables min max values are stored in a CSV file. Finally, we reduce our value range to -1 to 1. Map file generation and CubedSphere generation uses DLWP which in the background uses TempestRemap.

Algorithm for map file generation:

1. Firstly we will be needing an exodus file (extension '.g') for the input mesh and the output mesh. The GenerateMesh executables that come with TempestRemap can be used to achieve this.

2. Once the output and input meshes are generated, we need to generate the overlap mesh(i.e, the mesh obtained by placing the input and outpost mesh overtop one another and recalculating intersections). GenerateOverlapMesh executable can be used for this.

3. Once the overlap mesh is generated, we can generate the weight file, which holds the information on remapping one mesh to another. GenerateOfflineMap executable can be used to perform this.

The map file generated is stored in a specific location and is reused to save the computation time as it s expensive. After the map file is generated the ApplyOfflineMap executable will generate a remapped temporary file which in turn results in the CubedSphere generation with CubeSphereRemap() module imported from DLWP.remap.

4.3 Target variables

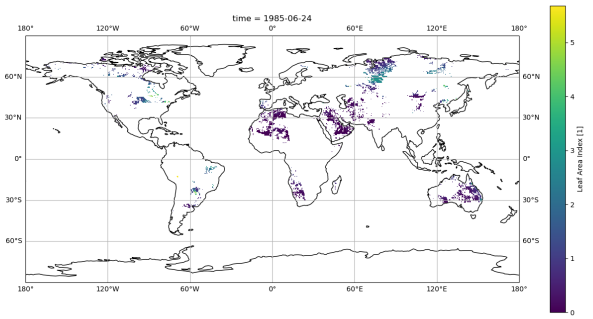
	Variable	Standard Name	Long Name	Units	Resolution
0	FAPAR	fraction_of_surface_downwelling_photosynthetic...	Fraction of Absorbed Photosynthetically Active...	1	(360, 720)
1	FWI	NaN	Fire Weather Index	NaN	(180, 360)
2	LAI	leaf_area_index	Leaf Area Index	1	(360, 720)
3	air	NaN	mean Daily Air temperature at 2 m	degK	(143, 144)
4	chlor_a	mass_concentration_of_chlorophyll_a_in_sea_water	Chlorophyll-a concentration in seawater (not l...	milligram m-3	(180, 360)
5	mlotst	ocean_mixed_layer_thickness_defined_by_sigma_t...	Density ocean mixed layer thickness	m	(180, 360)
6	precip	lwe_precipitation_rate	NOAA Climate Data Record (CDR) of Daily GPCP S...	mm/day	(143, 144)
7	so	sea_water_salinity	Salinity	1e-3	(180, 360)
8	sst	NaN	Daily Sea Surface Temperature	degC	(720, 1440)
9	zos	sea_surface_height_above_geoid	Sea surface height	m	(360, 720)

Table 4.3: Target variables

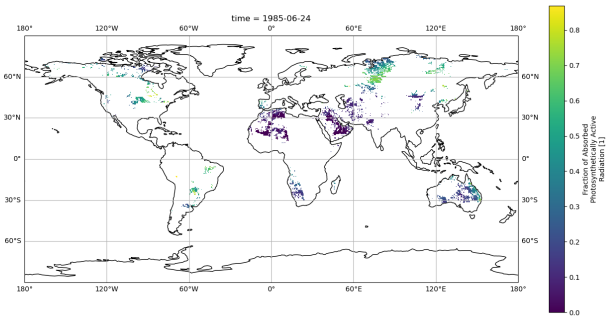
Variable	Data	Link
sst	NOAA data	hyperlink
precip	NOAA data	hyperlink
air	NOAA data	hyperlink
LAI and FAPAR	NOAA data	hyperlink
chlor_a	cds copernicus data	hyperlink
FWI	ECMWF	hyperlink
mlotst, so and zos	CMEMS	hyperlink

Table 4.4: Target variable data download links

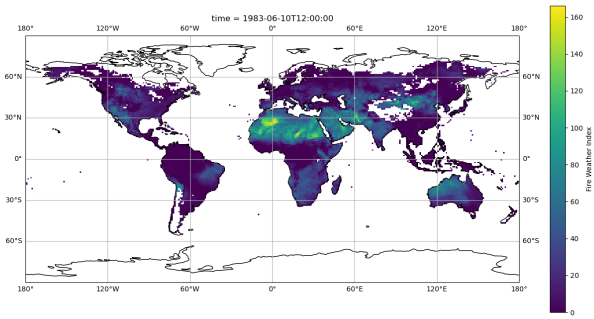
We have a total of 10 target variables. These are real observational data and forms the ground truth data for our deep learning algorithm. The ground truth data is global as well as daily data.



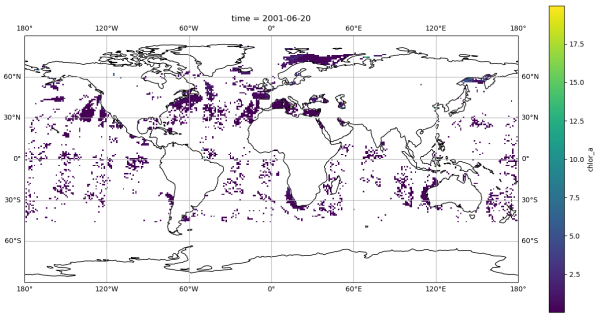
LAI



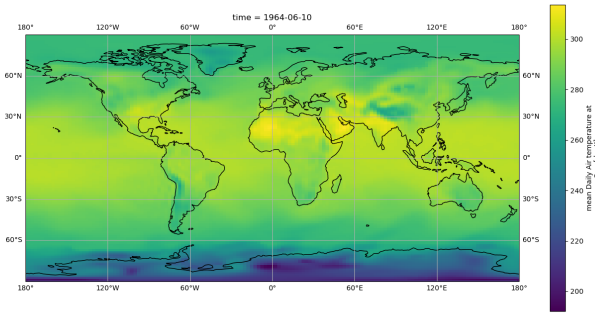
FAPAR



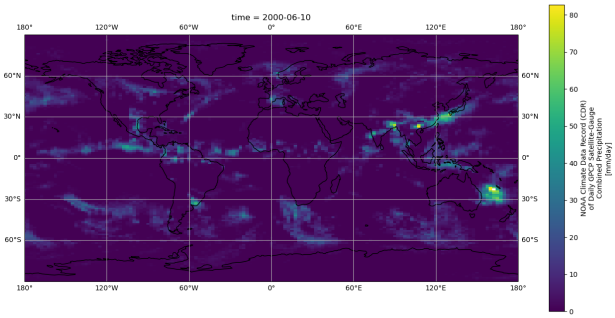
FWI



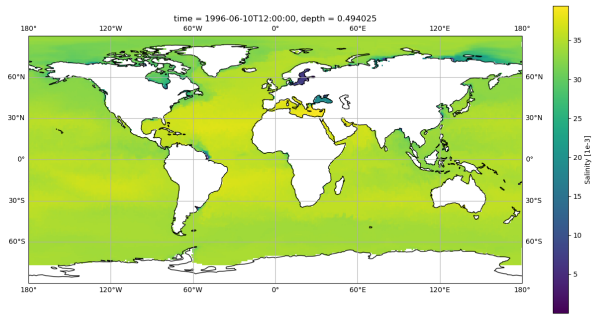
chlor_a



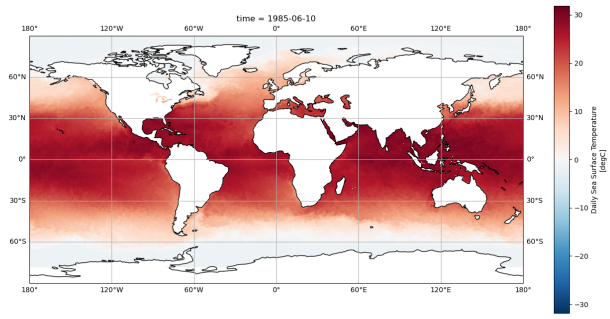
air



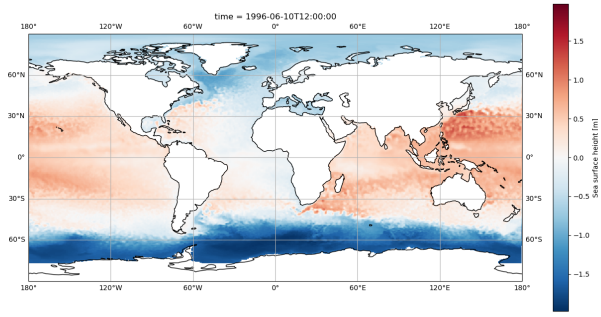
precip



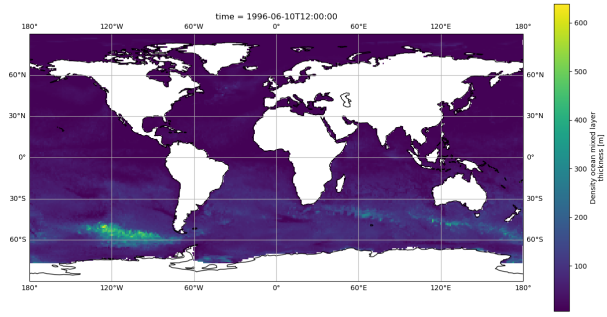
so



sst



zos



mlotst

Figure 4.6: Target Variables

Russia Africa

Figure 4.7: Russia and Africa in World map

4.4 Preprocessing Target Variables

As mentioned for input variables, the downloaded data is in batches of 10 years. But the target variables data is not in batches of 10. It is either in daily data in yearly files or as daily data itself. So there is a need to make our target variable data also batches of 10. Apart from this, all the preprocessing steps are done in the exact way we did for the input variable or model variables.

CubedSphere mapping minimizes the distortion caused by the spherical nature of the data. When the spherical dataset is transformed into CS, to reduce spherical distortions, we need to select the spatial resolution of CS faces. We went with resolution 96.

4.5 Why Cubed Sphere?

The most common and one of the most natural co-ordinate system for indexing data on the spherical globe is the general lat-lon (latitude-longitude) grid. Projecting 3D spherical earth to rectangle lat-lon grid itself has many problems. There can arise errors due to spherical conversion known as spherical distortion error. Also the lat-lon grid system has singularities at the north and south poles which makes the convolution operations on this grid difficult.

[Figure 4.7] shows Russia and Africa in the world map. This plot is in the general latitude-longitude grid format. Visually, Russia seems to be larger than Africa. But in the real scenario Russia is having 17.13 million km^2 area on the other hand Africa is having 30.37 million km^2 area. Thus stating the fact that Africa is 1.77 times as big as Russia. Visually, it contradicts this real fact. Thus, these lat-lon grid problem also affects the algorithms which runs on it. These are also due to spherical distortion error caused by the transformation from 3D spherical globe to

rectangular general lat-lon grid. Thus comes the importance of Cubed sphere transformation of this general lat-lon grid which is also discussed in [7].

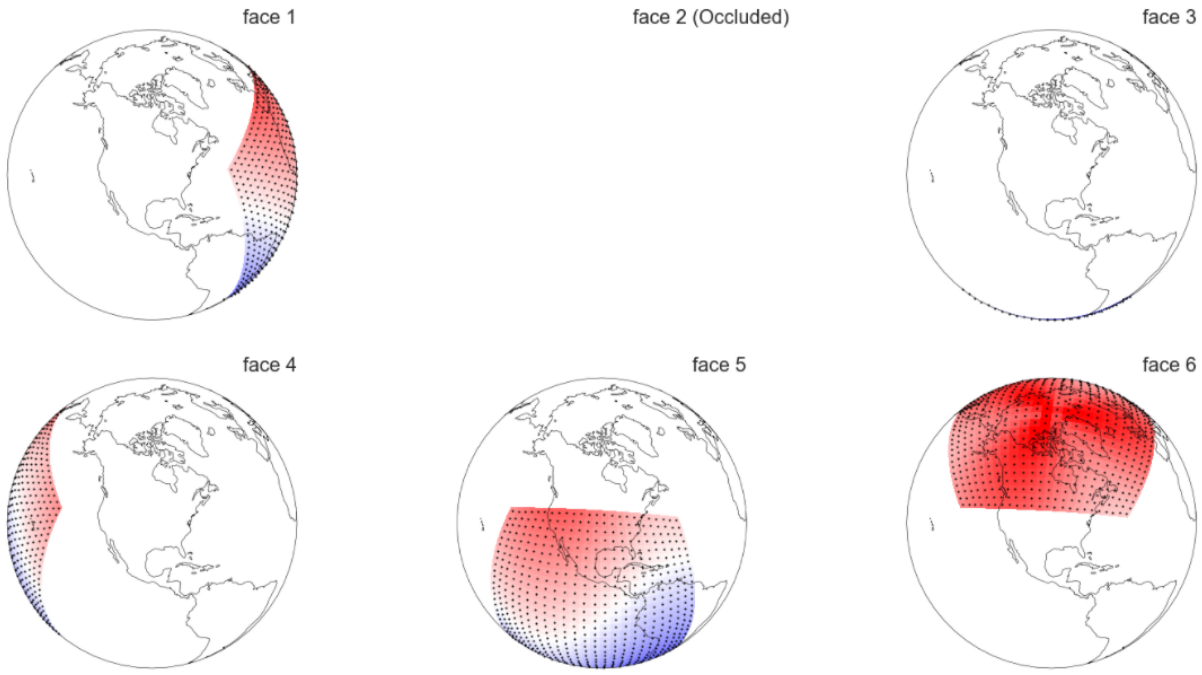


Figure 4.8: Cubed Sphere all 6 faces

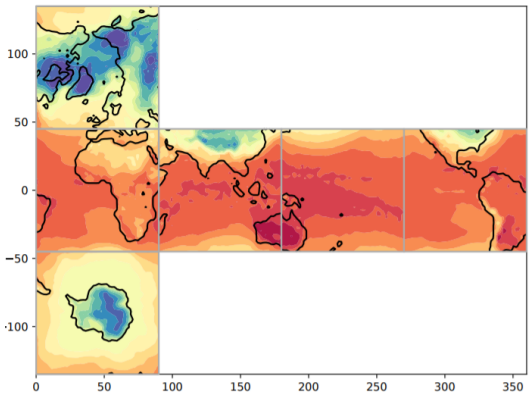


Figure 4.9: Flattened cubed sphere

CubedSphere considers the spherical globe to be a box with 6 faces. [Figure 4.8] shows all the 6 faces of the earth and how it look likes. Face 2 is occluded from the view as it faces the opposite direction. Usually the cubed sphere files will be with the dimensions ("face", "height", "width"). The height and width sets the grids inside each faces. As mentioned earlier we are taking height

and width both 96. Thus our dimensions will be (6, 96, 96) added with each time dimension making it spatio-temporal data. We can also change the resolution from 96 to a higher values. But the computation complexity will change vigorously. It will be so huge that our system might not be able to handle such computations.

[Figure 4.9] shows the flattened cubed sphere with the 6 faces. This is one of the many representation of flattened cubed sphere. This is able to capture the characteristics of spherical globe to a greater extent. Thus reducing or minimizing the spherical distortion error. This is why we are converting our data into cubedsphere with 6 faces and resolution as 96.

4.6 AI Model

The AI model used in this particular study is UNET. It is a CNN based model. It is first applied to biomedical images in 2015. The UNET model is able to localize and distinguish borders by doing classification on every pixel. The general architecture can be found in [Figure 4.10].

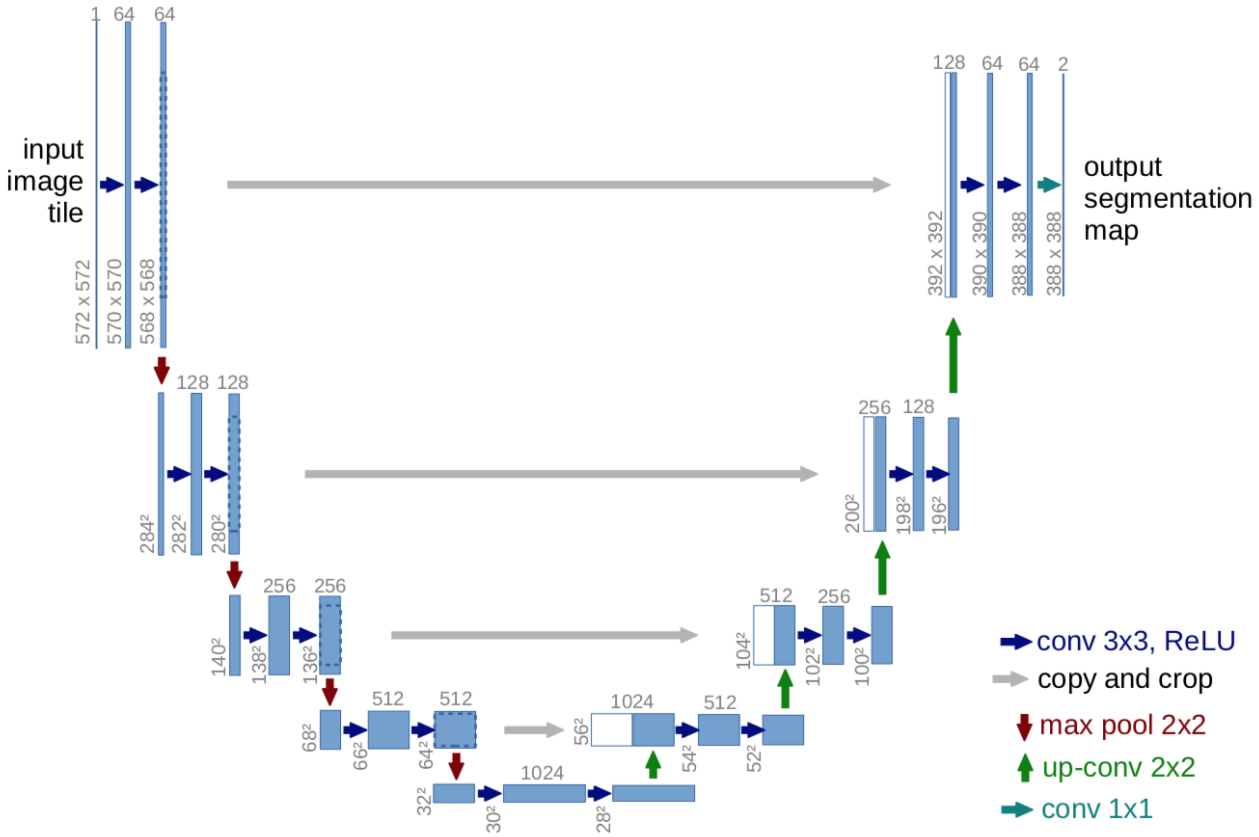


Figure 4.10: Unet general architecture

It has a 'U' shape. The architecture is symmetric and consists of a contracting path and a expansive path. The contracting path constitutes general convolutional process and the expansive path constitutes transposed 2d convolutional layers or simply upsampling. An alternate diagram can be found in [Figure 4.11] created for this particular study.

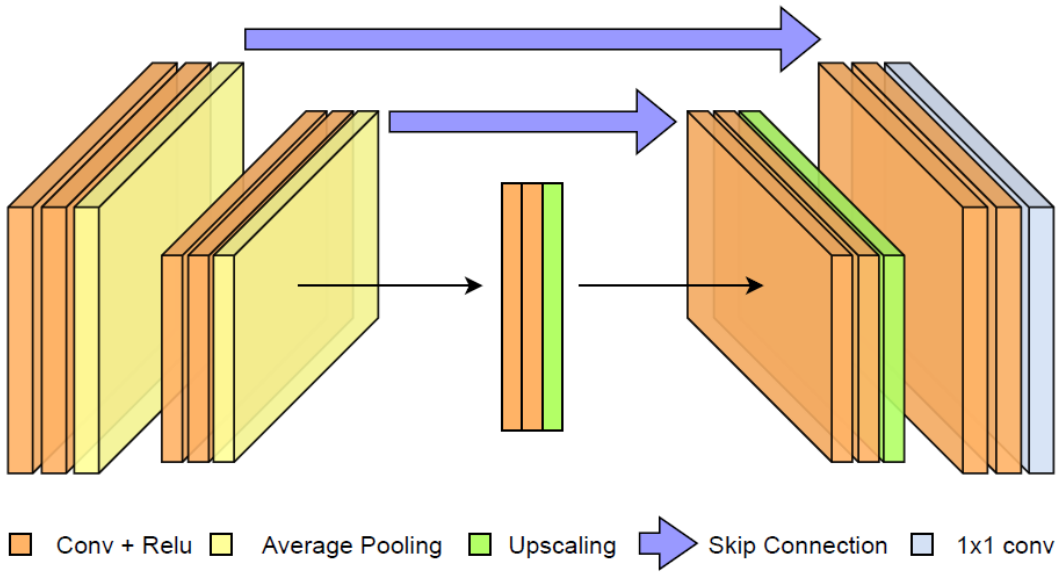


Figure 4.11: UNET

In [6], deep convolutional neural networks (DLWP) applied UNET to 500-hPa geopotential height reanalysis dataset. In the improved version, [7] (DLWP-CS) transformed the spherical global data into the CubedSphere and showed improved predictions rather than the general latitude-longitude datasets. The [1] discussed the use of modified DLWP-CS UNET architecture to predict the global precipitation forecast with good results. In this study, the optimizer used is adam optimizer and mse as the loss function.

Chapter 5

Surface Air Temperature - Methodology

One of the major problem faced by us as an earth inhabitant is the problem of global warming. Global warming is highly correlated to the increase in air temperature. This increase in air temperature affects human lives in a negative manner. It leads to change in climatic conditions such as rise in sea water level, global warming and growth of extreme events. People may suffer potential health problems if the air temperature is not on a suitable range. Thus forecasting air temperature is an important process in weather prediction as it affects human lives and properties.

5.1 Schematic

[Figure 5.1] shows the flow chart/the schematic of this study. As this is a univariate model, our predictor and predictand is both surface air temperature. The predictor, tas is a decadal hindcast daily product ranging from dates 1961-2026 from the source station IPSL-CM6A-LR and varient label r1i1p1f1. The predictand, air is a NCEP reanalysis daily average data ranging from 1961-2020. Both the predictor and predictand are global data only.
The data is then passed through the preprocessing stages mentioned in [chapter 4]. As a result Cubed Sphere data is generated for both the predictor and predictor. The resolution of the CubedSphere is 96. Thus, the dimension of the CubedSphere will be (6, 96, 96) corresponding to ("face", "height", "width"). Since this is decadal files, the final dimension will be ("time", "face", "height", "width") with "time" variable being either 3652 or 3653.
This cubedsphere date is our input to the AI model. UNET, being our AI model is a regression model in its background. Regression model is able to find the relationship among dependent variable and one or more independent variables. At it heart. this is a Supervised Learning

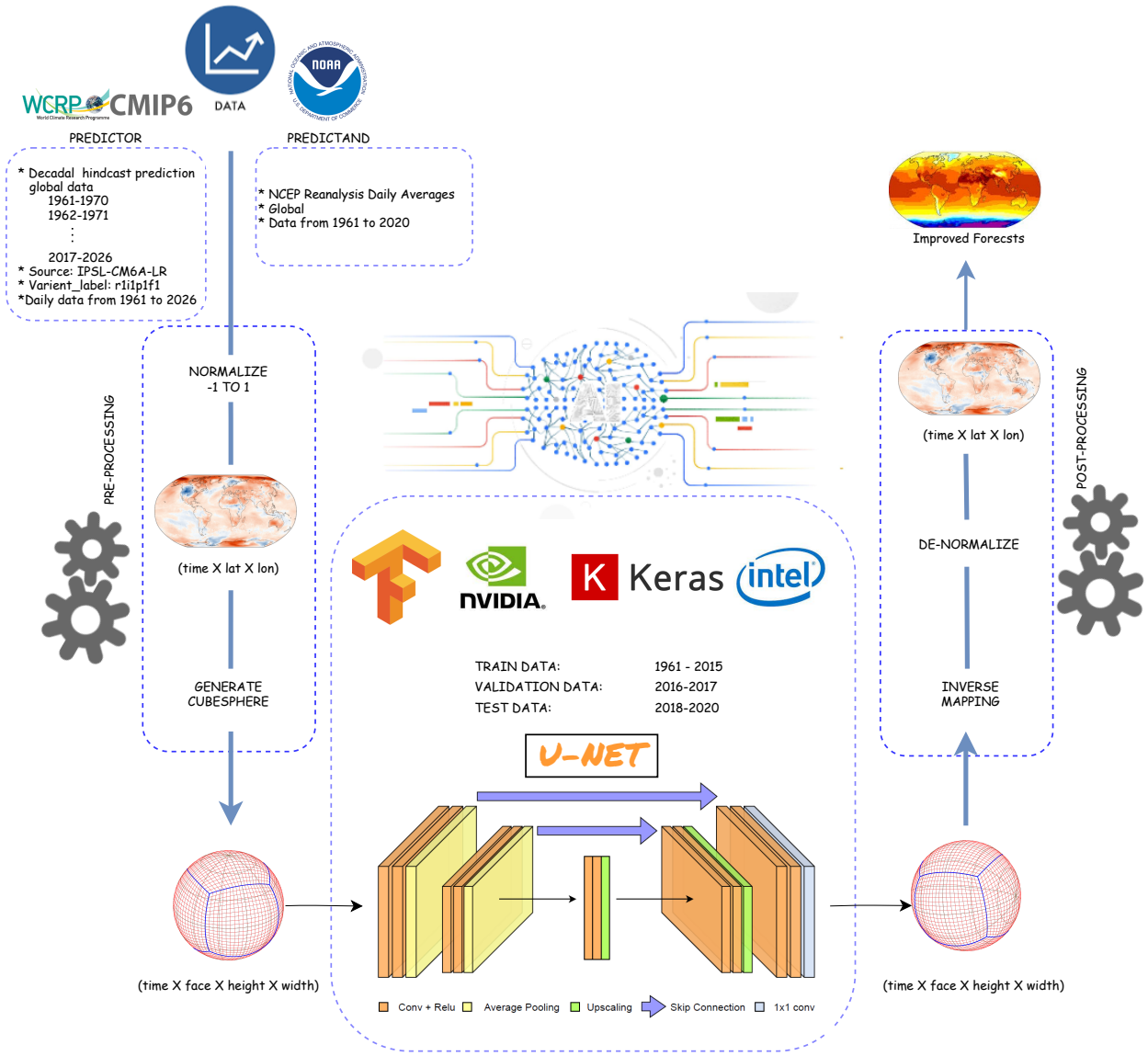


Figure 5.1: tas schematic

algorithm. Therefore we are splitting our data into train, validation and testing. Training set is a sample of data used to fit the machine learning model. Validation set is also a sample of the data which is used for the unbiased evaluation of the fit of the model with the train data by hyper tuning the parameters. Test set is the final unbiased evaluation on the model. In this study, we have taken the data from years 1961 to 2015 as our training data for the model. The dataset corresponding to 2016 and 2017 is taken as the validation data and the years from 2018 to 2020 are using as the test set for the evaluation of the model. The model is trained on a NVIDIA a100 40GB GPU. The batch size provided is 8. So for a year in training either time will be 3652 or 3653. Therefore total batches will be ceil(3652/8) = 457. That is 457 batches will be for a year. We have been running each year for 1000 epochs. On the specified GPU, it take 87 ms per step which makes 40 seconds per epoch. The model is getting trained in such a way that 1000 times the 1000 epochs per year will run. At each time the model weight files are being saved to a location. The training loss was around 0.0014 and validation loss being 0.0015.
The output from the model for the test set of data are also generated in CubedSphere format. Thus for evaluation it need to be converted back to regular latitude-longitude grid format. So it follows the Post processing stage, which is reverse tracing of the preprocessing stage. For de-normalizing we make use of the min-max file stored earlier and also the map files to reduce computation complexity.

5.2 Results

5.2.1 Australian Heat Wave 2019

"A heatwave in Australia occured in 2019 December with a record average temperature of 40.9°C on 17th. This was surpassed on 18th by an average air temperature of 41.9°C". The above is a news slice of what happened on Dec 2019 in Australia Heat Wave. We are trying to predict the above news using our AI model.
[Figure 5.2] and [Figure 5.3] represents the analysis done on the 17$^{\text{th}}$ and 18$^{\text{th}}$ December 2019 Australian heat wave events. air Observation is the Observational model output/Predictand. air Predicted is the output from our AI model and tas CMIP6 is the dynamical model output/predictor. The initial plots are the plots of Australia on those days. We can clearly see that our model output clearly outperforms the dynamical model output. The later plots are the bias plots of air predicted VS Observation and tas CMIP6 VS Observation respectively. The distribution plots used is typical box plots. The whiskers represent the minimum and maximum values of the distribution. The box represents the first quartile, median and third quartile respectively. This distribution helps

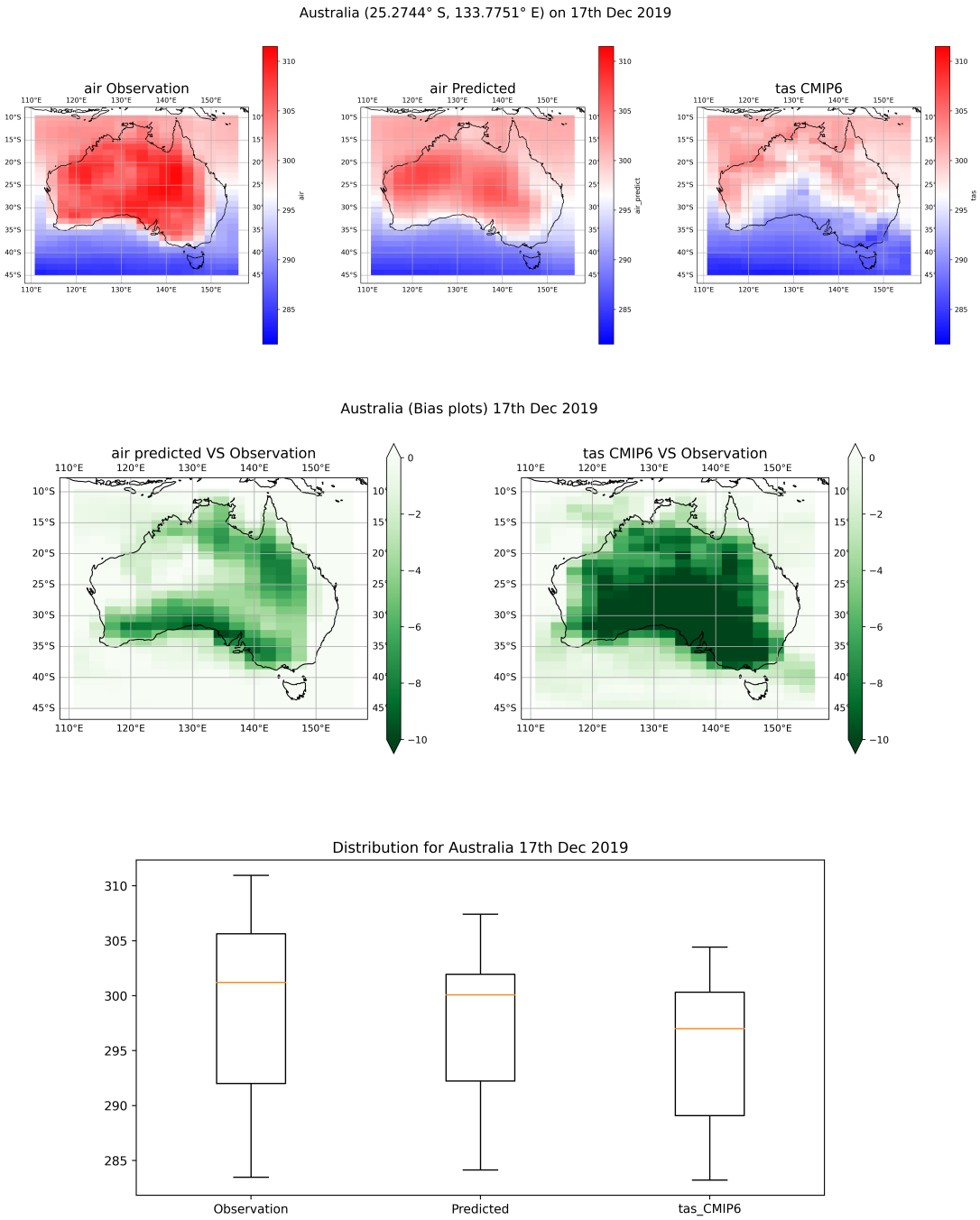


Figure 5.2: 17th DEC Australian heat wave

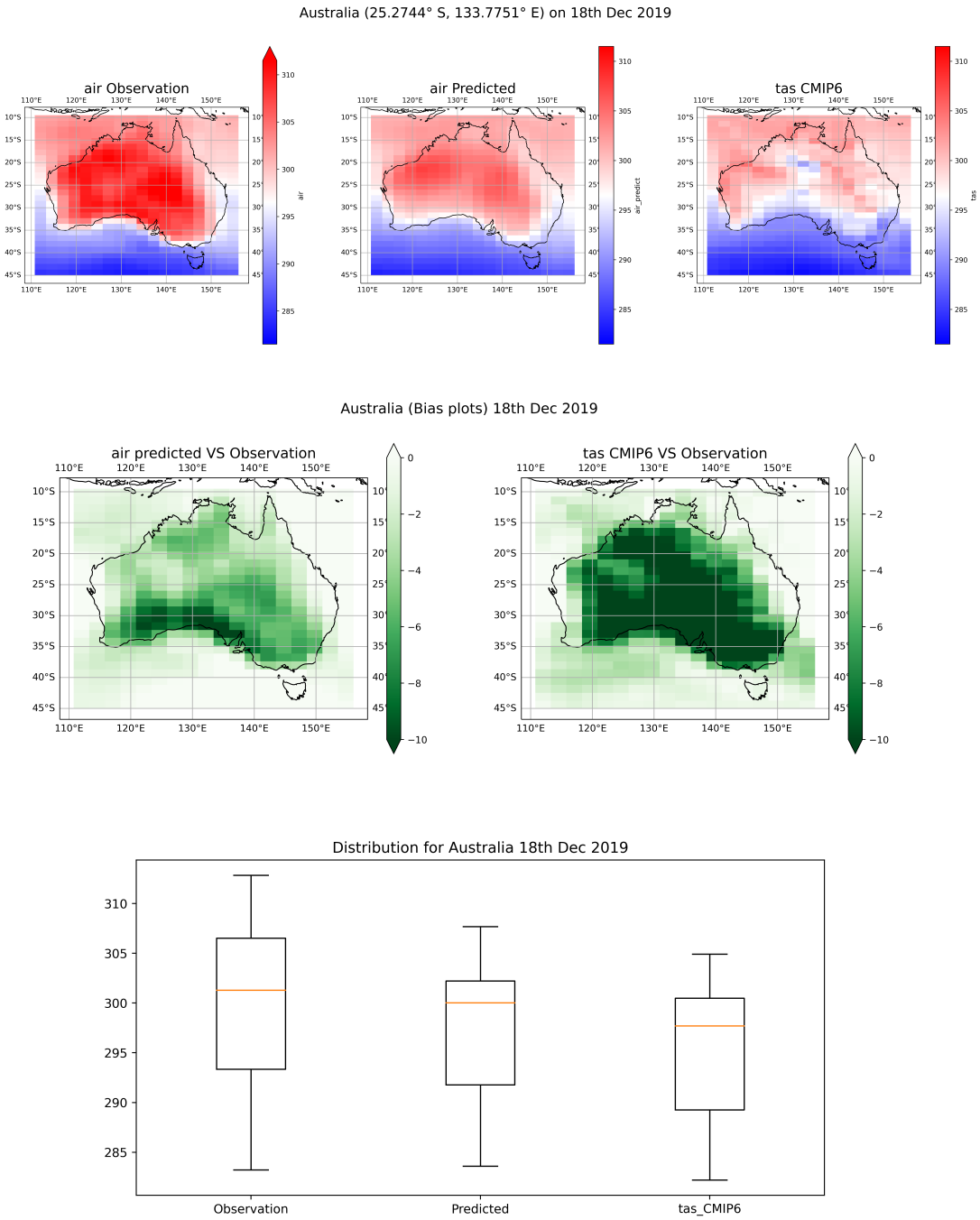


Figure 5.3: 18th DEC Australian heat wave

us to understand how our data is distributed, which is better than dynamical model distribution and more closer to Observation distribution.

		RMSE	MAE
17th Dec 2019	tas_CMIP6 VS Observation	6.554418551569722	4.499365197003986
	air_predicted VS Observation	3.250353195243999	2.2666073761785017
18th Dec 2019	tas_CMIP6 VS Observation	6.776259644971914	4.806816742084563
	air_predicted VS Observation	3.5984313105937176	2.6706174205069684

Table 5.1: Evaluating Australian heat wave

The evaluation metrices showed in [Table 5.1] shows that our model performs way better than the dynamical model for predicting the 2019 Australian heat wave.

5.2.2 European Heat Wave 2019

In late June 2019, there was distinct European heat waves, which set all time high temperature records in Belgium, France, Germany, Luxembourg, Netherlands and the United Kingdom. It resulted in the deaths of 567 people and according to scientists it was caused by high pressure and winds from the Sahara Desert affecting larger portion of the continent.
The June European heat wave started from 24 June and lasted till 02 July 2019. Thus for the calculation purpose the grid-wise mean data is taken for the above mentioned dates. The results are presented in [Figure 5.4].

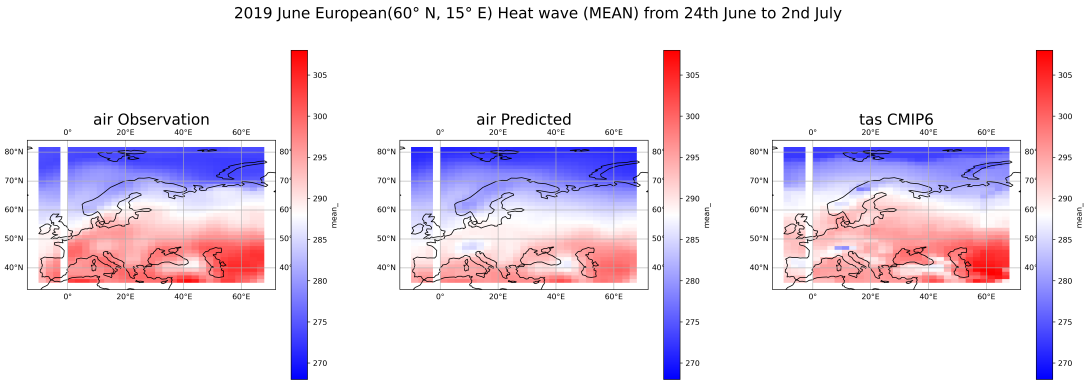


Figure 5.4: European heat wave 2019 June

		RMSE	MAE
Late June 2019	tas_CMIP6 VS Observation	3.3099693040509277	2.5882628302153847
	air_predicted VS Observation	3.319922285432378	2.602152770028401

Table 5.2: Evaluating European heat wave

It can be seen that our predicted model output is almost similar with the dynamical model output which is presented in the [Table 5.2].

5.2.3 Maine Heat Wave 2020

Caribou, Maine tied its all-time record high of 36°C on June 19, 2020. The heat wave, combined with abnormally dry conditions, led to numerous forest fires in the area.

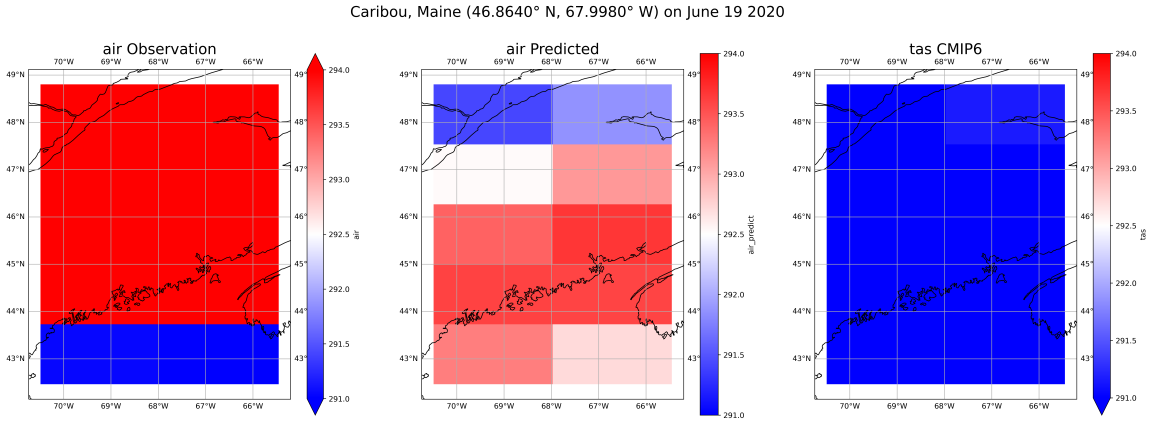


Figure 5.5: Manine heat wave

It is visually clear that our model output is better than the dynamical model output from the [Figure 5.5]. Even the RMSE and MAE metrics also favours our model predicted output as in [Table 5.3].

		RMSE	MAE
Maine June 2020	tas_CMIP6 VS Observation	6.378521914369058	5.856295530749435
	air_predicted VS Observation	3.7410388813136333	3.5182634151078958

Table 5.3: Evaluating Maine heat wave

Bibliography

[1] Manmeet Singh, Bipin Kumar, Suryachandra Rao, Sukhpal Singh Gill, Rajib Chattopadhyay, Ravi S Nanjundiah, Dev Niyogi. Deep learning for improved global precipitation in numerical weather prediction systems. arXiv.org, Physics, Atmospheric and Oceanic Physics (physics.ao-ph); Machine Learning (cs.LG), arXiv:2106.12045v2 [physics.ao-ph], 24 Aug 2021.

[2] Schiermeier, Q. (2010). The real holes in climate science. Nature News, 463(7279), 284-287

[3] Jonathan A.Weyn, Dale R.Durran, Rich Caruana. Improving Data-Driven Global Weather Prediction Using Deep Convolutional Neural Networks on a Cubed Sphere. arXiv:2003.11927v1 [physics.ao-ph], 15 Mar 2020.

[4] Evaluation of ENSO simulations in CMIP5 models: A new perspective based on percolation phase transition in complex networks.(https://www.nature.com/articles/s41598-018-33340-y.pdf)

[5] Kumar, P.; Kashyap, P.; Ali, J. Temperature Forecasting using Artificial Neural Networks (ANN). J. Hill Agric. 2013.

[6] Weyn, J.A., Durran, D.R. and Caruana, R., 2019. Can machines learn to predict weather? Using deep learning to predict gridded 500-hPa geopotential height from historical weather data. Journal of Advances in Modeling Earth Systems, 11(8), pp.2680-2693.

[7] Weyn, J.A., Durran, D.R. and Caruana, R., 2020. Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. arXiv preprint arXiv:2003.11927

[8] Chattopadhyay, S.; Jhajharia, D.; Chattopadhyay, G. Univariate modelling of monthly maximum temperature time series over northeast India: Neural network versus Yule-Walker equation based approach. Meteorol. Appl. 2011, 18, 70–82.

[9] Zhang, Z.; Dong, Y.; Yuan, Y. Temperature Forecasting via Convolutional Recurrent Neural Networks Based on Time-Series Data. Complexity 2020, 2020.

[10] Li, C.; Zhang, Y.; Zhao, G. Deep Learning with Long Short-Term Memory Networks for Air Temperature Predictions. In Proceedings of the 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), Dublin, Ireland, 16–18 October 2019; pp. 243–249.

[11] Smith, B.A.; Hoogenboom, G.; McClendon, R.W. Artificial neural networks for automated year-round temperature prediction. Comput. Electron. Agric. 2009, 68, 52–61.

[12] Akram, M.; El, C. Sequence to Sequence Weather Forecasting with Long Short-Term Memory Recurrent Neural Networks. Int. J. Comput. Appl. 2016, 143, 7–11.

[13] Sundaram, M.; Prakash, M.; Surenther, I.; Balaji, N.V.; Kannimuthu, S. Weather Forecasting using Machine Learning Techniques. Test Eng. Manag. 2020, 83, 15264–15273.

[14] Kreuzer, D.; Munz, M.; Schlüter, S. Short-term temperature forecasts using a convolutional neural network — An application to different weather stations in Germany. Mach. Learn. with Appl. 2020, 2, 100007.

[15] Lee, S.; Lee, Y.S.; Son, Y. Forecasting daily temperatures with different time interval data using deep neural networks. Appl. Sci. 2020, 10, 1609.