Satellite nowcasting of cloud coverage via machine learning

DINA LAZORENKO, University of Twente, The Netherlands

Additional Key Words and Phrases: image-to-image prediction, cloud mask, satellite images, cloud coverage, grid data, geostationary position, Mercator position, forecasting

ABSTRACT

Satellites orbiting the earth make it possible to provide increasingly accurate weather forecasts. Through observations and numerical weather predictions, meteorologists can make near-future weather forecasts which have a major impact on everyday decisions in different societal sectors. However, unexpected cloud appearances often lead to inaccuracies in nowcasting weather or radiation. We propose the usage of machine learning via a Convolutional LSTM Neural Network. In particular, we validate the performance of learning spatial and temporal patterns within a sequence of satellite images using a Convolutional Long Short-Term Memory model architecture for cloud coverage prediction. The accuracy of the machine learning model was found to be 90 percent upon training on a dataset spanning a period of two years. However, additional efforts are deemed necessary to address and remove biases present within the model.

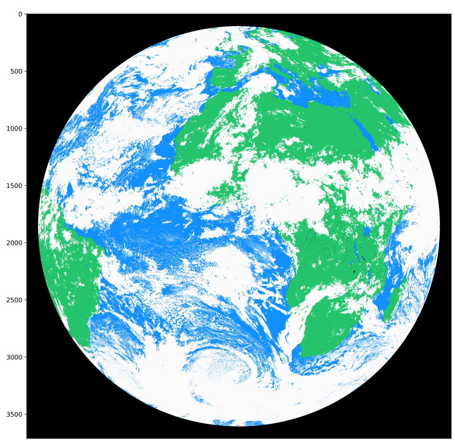


Fig. 1. A scene of the cloud mask of the earth on geostationary projection on a pixel level. Each pixel is classified as either clear sky over water(blue), clear sky over land (green), cloud (white) or no data. [19]

1 INTRODUCTION

In meteorology, an optimal forecast refers to a prediction of the future state of the atmosphere that exhibits the least deviation from actual weather observations. This is accomplished through the application of state-of-the-art numerical models and statistical methods with the objective of attaining the greatest degree of accuracy possible.

This research is carried out under the supervision of Infoplaza B.V, a weather forecast service provider located in Houten, Netherlands. Currently, Infoplaza does not have a forecasting system that incorporates observations to predict future development in short time intervals. As, a result the company is working towards the development of a satellite-based *nowcast* or *short-term* cloud mask forecast.

While weather models are utilized for long-term forecasts covering a period from a few hours to several days, the absence of a cloud nowcast within the company requires the addition of a machine learning model. This will be utilized to adjust the first hours of the so-called "optimal" forecast, which is based on various weather models and post-processing steps.

This research focuses on whether it is possible to make a model which can take a series of satellite images of cloud coverage from gridded data files in `GRIB` format provided by EUMETSAT as input and predict a new image. The new images show a prediction of cloud distribution for the area from a series of past images.

EUMETSAT, the meteorological agency of Europe has several satellites located above the earth which provide meteorologists with satellite images [4].

In this research we make use of satellite images at different frequencies created by the *Seviri* instrument of the satellite. The satellite providing us with images belongs to the Second Generation of Meteosat Satellites whose satellites, specifically Meteosat 9, 10 and 11, can produce data for weather forecasts.

The satellite providing us a datset of satellite images is in geosynchronous orbit, matching the rotation speed of the earth. The *zero degree service* [14] is the main mission of EUMETSAT which provides stable satellite images over a given region where the satellite is orbiting directly above the equator.

Measurements observed on images of satellites can help detect different types of objects reflecting light differently. Thus, cloud objects and even a whole cloud mask over an area can be identified. The main purpose of cloud mask images is to differentiate between cloudy and cloud free surfaces on a given area of the earth.

1.1 Research Question

How can a deep learning model forecast cloud coverage using a series of satellite images?

The following sub questions should help to answer this question:

(1) How does the produced data from meteorological satellites covering Europe look like?
(2) Which neural network architectures are used for image-to-image prediction?
(3) How effective is ConvLSTM for cloud coverage nowcasting?

Author's address: Dina Lazorenko, d.lazorenko@student.utwente.nl, University of Twente, P.O. Box 109, Enschede, The Netherlands, 7521EB.

2 RELATED WORKS

In this section, we describe the state-of-the-art in nowcasting of cloud coverage. We found research in the field of nowcasting where machine learning models are used in order to predict weather phenomena. However, the majority of these studies have been conducted between 2015 and 2022, which indicates that the application of machine learning in nowcasting is a relatively new area of research. The overarching goal of these studies is to improve the accuracy of weather forecasting in order to facilitate daily life and protect individuals by providing accurate predictions of near-future weather conditions.

The currently most successful and well-known research into weather nowcasting by applying machine learning is the Deep Generative Model of Radar (DGMR) for precipitation nowcasting by DeepMind.

Highly modern nowcasting methods used by meteorologists have problems capturing unexpected weather phenomena.

DeepMind is a British research laboratory acquired by Google which developed a machine learning model for probabilistic precipitation nowcasting from radar data which allows to give a more accurate and non-linear prediction of precipitation. DeepMind makes use of a machine learning architecture called `GAN` (Generative Adversarial Network) which is a type of architecture used for DGMR.

The DeepMind `GAN` model architecture consists of the *generator* and two *discriminators*. As in other `GAN` models the discriminators need to predict if a given image is real or fake. The generator competes against the two discriminators to produce a convincing enough image to be accepted as real.

Both the generator and discriminators make use of multiple convolutional and LSTM layers within their architecture such as `ConvGRU`, `SNConv2D`, `SNConv3D` and more [31].

The utilisation of this machine learning architecture has resulted in a demonstration of an accuracy of 88 percent. This model is able to produce a nowcast with a lead time between 5-90 minutes on a 1536 by 1280 kilometer area. DeepMind is using a dataset of three years where radar images have been collected in a time interval of five minutes. The radar images represent a gridded dataset where each grid cell corresponding to the pixel cell in a radar image represents an area of 1km by 1km. In order to provide valuable results for precipitation nowcasting, DeepMind also uses data from satellite images, more precisely DeepMind makes use of a `MetNET` algorithm which relies on several layers of input data. This means that in addition to radar data, satellite images from the geostationary operational environmental satellite 16 (GEOS 16) are used. The dataset used by DeepMind is limited to the area of the United Kingdom [35].

The paper "Ten-minute prediction of solar irradiance based on cloud detection and a long short-term memory (LSTM) model" discusses the use of a long short-term memory (LSTM) model to predict solar irradiance with a ten minute time horizon. The authors first use a cloud detection algorithm to create a cloud mask. The resulting image is a binary cloud mask where pixels representing clouds are labled as `1` and pixel representing clear sky are labeled as `0`. The cloud mask, the Global Horizontal Irradiance (GHI) and key meteorological factors influencing the GHI are passed to an LSTM model. This LSTM model makes use of the sigmoid and tanh activation functions. The paper reports that by using a dataset of solar irradiance and cloud coverage information as input to the LSTM model, the model is able to make more accurate predictions of solar irradiance compared to using only historical solar irradiance data[46].

"Convolutional Networks for Biomedical image segmentation" is a research paper that presents a deep learning model for segmenting biomedical images via a machine learning model architecture called U-Net. The model is based on a fully convolutional neural network architecture, and it utilizes a symmetrical, concatenating structure to combine information from different resolution scales in the image. The authors demonstrate the effectiveness of U-Net on various medical image segmentation tasks and show that it outperforms existing methods in terms of accuracy and robustness[37].

Further examples of research have utilized machine learning architectures for forecasting hazards such as landslides and thunderstorms such as in the papers "New Potential for Local and Regional Deep-Seated Landslide Nowcasting"[41] and "Nowcasting Thunderstorm Hazards Using Machine Learning: The impact of Data Sources and Performance"[29].

These studies demonstrate the potential of machine learning techniques for accurately predicting these hazards.

3 METHODOLOGY

3.1 Literature and Business Understanding

The motivation for this research is to utilize the knowledge produced by this research to create and deploy a deep learning model capable of nowcasting cloud coverage. The prediction of cloud coverage can be used to enhance existing weather applications or as a input to predict energy production, among other uses [46].

The aim of the research is to enhance cloud coverage prediction accuracy through the use of machine learning models trained on a large volume of satellite image data. The models will identify the exact location and distinguish different cloud types in the images. The machine learning model is intended to complement the existing *Adjust to Obs* processing step, which adjusts the initial hours of the optimal forecast using observations such as temperature, humidity and speed.

Furthermore, this research aims to improve the prediction from photovoltaic cells by utilizing the results from the obtained cloud coverage nowcast. This information is expected to enhance the accuracy of the weather apps provided by Infoplaza by displaying images of cloud mask prediction.

Fig. 2. An image obtained from the Drops weather app by Infoplaza. The image depicts the forecasted cloud coverage over Europe. It offers a visual representation of the expected distribution of clouds across the region, providing valuable information on the anticipated weather conditions. [3]

3.2 Applied Methodology during pre-processing

The task of organizing and managing large amounts of satellite images from multiple sources is a complex and challenging endavour. To address this issue, we utilize the Extract, Transform and Load (ETL) methodology[21]. We consolidate compressed satellite images from different directories into a unified directory.

1. Extract

Firstly, we extract source data (`GRIB` files) from the EUMETSAT website and transfer this data to an external server, where `GRIB` files are extracted for the transformation stage. Our goal is to temporarily store data in a staging area to restart the ETL process if required. Using data chunks helps us to avoid resource limitations and increase pre-processing efficiency.

2. Filter

The next step after extracting data from zip files is to transform the acquired data, including filtering to a list of only GRIB files and excluding metadata files.

3. Transform

After completion of filtering and sorting, we determined that no further cleaning was required, as the data was free of broken, duplicate or missing records. We executed the reprojection of the GRIB files successfully without deviations. The ideal duration of reprojection for monthly data was approximately 2.5 hours.

4. Load

The final step in the ETL process involves loading the data into the destination directory. Each reprojected file is promptly loaded into the destination folder during the transformation phase.

3.3 Expert Validation Methodology

To ensure the accuracy of input data for our model, we employed an expert validation methodology. Five experts in the field of meteorology and data technology were presented with a set of satellite images in geostationary and Mercator reprojection. The experts were given the task to classify the images according to the specified questions. The results indicated that the experts achieved a high level agreement in their answers, with only minimal deviations. This agreement supports the conclusion that the reprojection from geostationary to Mercator was carried out successfully[27] see figure 9.

3.4 Machine Learning Architecture

In this research, we utilize a ConvLSTM model for forecasting cloud coverage. The ConvLSTM is based on a Long Short Term Memory (LSTM) architecture, which uses both long term and short term memory paths for predictions. LSTM was designed to address the vanishing and exploding gradient problem, enabling the model to improve its weights incrementally during training. However, we selected the ConvLSTM over the LSTM because the input data for each cell of the LSTM model is an image. The ConvLSTM preserves pixel relationships in these images through its use of convolutional operations.

We start off with a base model which we iterate upon during the experiments to explore improvements to this model. The base model consists of two layers. The first layer is a Convolutional LSTM and the second layer is a Convolutional layer.

The first layer is configured to use `20` filters of dimensions `(5, 5)`, since we are interested in a next-frame prediction we only use the final short-term memory, which has the dimensions `(800, 800, 20)`. In the first layer we also switch the *tanh* activation function with a *ReLU* function, this is because our output classes range from `[0, 3]`.

The output of the first layer is passed to the second layer which is a 2D convolutional layer, it treats the 20 images produced by the previous layer as different channels. The convolutional layer then uses 4 filters to produce a result of dimensions `(800, 800, 4)`. The 4 values in each pixel correspond to probabilities of the pixel belonging to one of the possible cloud mask classes (See table 1).

The model is compiled with a sparse categorical crossentropy and adam optimizer. We utilize sparse categorical crossentropy as a loss function because our classes are represented by integers and are not one-hot encoded. The adam optimizer was used based on other models of the same task [30].

In detail the model can be formulated by the following equations:

$$f_t = \sigma(W_{xi} * X_t + W_{hi} * h_{t-1} + b_i)$$
$$i_t = \sigma(W_{xf} * X_t + W_{hf} * h_{t-1} + b_f)$$
$$o_t = \sigma(W_{xo} * X_t + W_{ho} * h_{t-1} + b_o)$$
$$C_t = f_t \odot C_{t-1} + i_t \odot ReLU(W_{xc} * X_t + W_{hc} * h_{t-1} + b_c)$$
$$h_t = o_t \odot ReLU(C_t)$$

f_t, i_t, o_t, correspond to the forget gate, input gate and output gate. C_t is the current cell state and h_t is the current short term memory.

These equations come from the paper that introduced ConvLSTM [38], except that we replaced *tanh* with *relu*.

The *sigmoid* function is a mathematical function which maps any input value to a value between zero and one. By taking the values between zero and one. In a LSTM cell sigmoid is used for example to determine the percentage of values to keep in memory.

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

On the other hand the *ReLU* (Rectified Linear Unit) activation function gives the maximum between 0 and the provided input.

$$Relu(x) = max(0, x)$$

Since we are predicting the next image in time, we cannot expect negative output results as predictions by our machine learning model.

The ConvLSTM model was firstly introduced by [38] in order to solve a precipitation nowcasting problem. Using a ConvLSTM we allow the model to learn features which are not only dependent on the individual pixel values as well as on the pixel values of the surrounding area by performing convolutional operations. In our convolutional layer we are making use of a kernel which works as a filter sliding over the image and performing a mathematical operation on each pixel value. The results of the mathematical operations create a new filtered image. This image is then used as an input for the next layer of the model. The model learns patterns of the cloud mask through convolutional operations. Thus, we can keep spatiotemporal pixel relationships when performing nowcasting (See Appendix C, B).

4 EXPERIMENTAL SETUP

4.1 Dataset

The dataset consists of satellite images taken over a two-year period by EUMETSAT, the European Organization for the Exploitation of Meteorological Satellites, which supports weather forecasting and other meteorological and climatological applications.

The images were taken every 15 minutes and range from `1000` to `2000` per month. The satellite images are stored in `GRIB` (GRIdded Binary) format, a binary encoded weather data storage format defined by the World Meteorological Organization (WMO) and used by EUMETSAT in accordance with the GRIB2 specification[7].

Data inside a GRIB file is stored in records and data in records may contain different information depending on the meteorological use case. In our case a GRIB file contains one record. The GRIB file has a cloud mask resolution of `3712x3712` pixel for single geostationary image of the earth. The gridded data displays the satellite imagery and each data point is described by its longitude and latitude.

The stored data also includes metadata about the satellite image, such as the height of the satellite above the earth and the projection type, which are used for reprojecting the images. Each pixel in the image can belong to one of the classes in table 1, in practice however the pixels only belong to the following classes:

(1) `clear over water`
(2) `clear over land`
(3) `cloud`

Each grid cell, represented by a pixel, has a resolution size of 3km by 3km at the sub-satellite point which is the point directly above the satellite when drawn as a line to the centre of the earth. The pixel size increases with distance from the sub-point, meaning a pixel contains more information in the middle of the image than at the edges (See figure 10).

4.2 Experiments

We have conducted a set of experiments in order to analyse the best parameters to be used with our model. All experiments are based on the model architecture described in the section 3.4 Machine Learning Architecture, each including one change. We executed each experiment twice. Firstly, we trained the model on a dataset of one month. Secondly, with an extension of resources we trained the model on a dataset of two years.

4.2.1 Experiment 1. In accordance with the theoretical specification of our `ConvLSTM` model, we carried out an initial experiment utilizing the `Keras` implementation of the `ConvLSTM2D` layer.

The model was constructed using a sequential architecture with the `ConvLSTM2D` layer serving as the initial layer. This layer receives a block of 5 satellite images, each of size 800 by 800 pixels. The `'return_sequence'` parameter was set to false, in order to only return the last 20 images, which were generated based on the previously trained outputs of the previous and current cells.

The first layer was followed by a Convolutional layer, which reduced the number of images to the number of 4 classes via the use of filters.

The `'from_logits'` parameter was set to false while compiling the model, in order to apply the *softmax* activation function by default. The performance of the model was evaluated through the use of `SparseCategoricalAccuracy` class which gives the count of correct pixels over the total, this metric applies *argmax* of probabilities to find the final class and compares the predicted class to the ground truth. This experiment was executed for one epoch.

4.2.2 Experiment 2. The following experiment we intend to analyze the existence of overfitting by increasing the number of training iterations by setting the number of epochs to 3. From the examination of the validation accuracy, it appears that the model has ceased to learn effectively after the initial epoch. This is evidenced by the minimal increase in accuracy observed between the first and subsequent epochs, with a fluctuation of approximately `0.0003` and `0.0007`. Based on the aforementioned observations, we can infer that overfitting is not present in the current model, which comprises approximately `42080` parameters.

4.2.3 Experiment 3. In this experiment, we simplified the labeling to distinguish between the presence and absence of clouds, with the desired outcome being an an image that depicts only the areas containing clouds or clear sky. We do not take the sea and land areas into consideration. To account for this binary categorization, we use the Binary Crossentropy loss function instead of the Sparse Categorical Cross Entropy Loss function previously used. Additionally, we altered the filter size in the final layer to the value of one. We are using the *sigmoid* activation function instead of the *relu.*

4.2.4 Experiment 4. In this experiment, a dropout layer is applied after the first and second layers. During each iteration, the dropout layer sets 20 percent of the outputs from the previous layer to zero, creating a different set of outputs with each iteration. This is done to prevent overfitting by reducing potentially existing dependencies of the model on any features. The purpose of this experiment is to evaluate whether the model relies on memorization to make its predictions, and to assess its overall performance.

4.3 Validation metrics

To assess the performance of our model, we determine the pixel accuracy utilizing both training and validation sets. Given that the validation dataset was not utilized during the training process and our model architecture selection was not contingent upon the validation dataset performance, it is not required to utilize a testing dataset and the validation dataset alone is sufficient for the purpose of evaluating our model.

To evaluate the performance of our machine learning model, we use the accuracy derived from the sparse categorical cross entropy. The sparse categorical cross entropy is a loss function applied during the training process, which quantifies the discrepancy between the predicted class probabilities and the actual class labels. The validation results obtained through utilization of the sparse categorical cross entropy can be found in the results and discussion section 5.

We conducted a confusion matrix analysis on a single image with binary predictions. The aim of this analysis is to evaluate the performance of our model in classifying the pixel values as cloud or non-cloud. The results of the confusion matrix can be found in the results and discussion section5.

$$ioU = \frac{truepositive}{truepositive + falsepositive + falsenegative}$$

4.4 Implementation Details

4.4.1 Resources. The available resources were used to complete the project: A server with `16 CPU cores, 32 GB RAM and 360GB disk space`. The data was initially uploaded to the server by Infoplaza and an account was created for the `jupyterlab` development environment. The `jupyterlab` user interface was used to interact with the data in order to create the machine learning model.

4.4.2 Used packages. In order to ensure that all needed functionalities are performed, we have made use of the following packages and modules: (Overview: 2).

4.4.3 Pyresample. Pyresample is a package designed to provide functions for working with geospatial data in Python. We are making use of Pyresample in order to transform the satellite images from geostationary projection to mercator projection.

4.4.4 Reprojection. In order to be able to visualise a two dimensional image of the future cloud distribution of a given area on earth map reprojection needs to be applied. Map reprojection is a set of functions f which convert the geographic coordinates from the spherical earth into a cartesian coordinate system with points x and y.

Different projections are used depending on the target reason for usage. Within the framework of this project the use of mercator was chosen for the given business case in order to allow the potential use of images in web based applications such as the apps of Infoplaza.

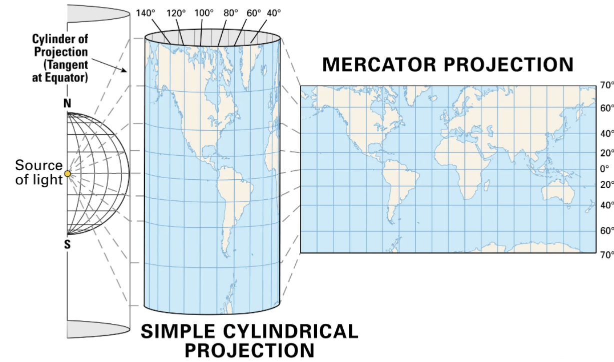


Fig. 3. The following image displays the creation of a Mercator reprojection depicting the process with a visual illustration. The image depicts a piece of paper being wrapped around the globe and then unfolded into the Mercator map, which serves as the input for our machine learning model.

4.5 Generator

In order to make use of RAM efficiently and avoid crashing of the server on which experiments were run, we made use of a Generator to generate batches for training and validation while only loading into memory the required cloud mask images.

The generator produces features and labels arrays for each batch. For the purposes of training our model our generator can be configured to produce batches with a given batch size and block size. The block size indicates the number of sequential images that are fed to the model before the label.

This generator was implemented by extending the `tf.keras.utils.Sequence` class and implementing the required methods.

4.5.1 Sorting Images in Chronological Order. In order to predict the next image of cloud coverage in time we need to provide our model with a series of satellite images in time. To ensure that only numpy files remain in the first a filter is applied to remove files not in numpy format. Afterwards the files can be sorted in chronological order to blocks of time series.

Time series are used to predict the next image in sequence because they allow us to model the temporal dependencies between the previous images.

Converting files through their metadata to datetime objects facilitates comparison between the files because a datetime object includes built-in comparison operators.

5 RESULTS AND DISCUSSION

5.1 Experiment 1

Results by training the model on a dataset of one month:

Epoch Number	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.7169	0.6867	0.6496	0.7222

This initial experiment demonstrates that, despite achieving an accuracy of 72 percent with a limited amount of data, there is still a significant level of loss present, which is nearly equivalent to the validation accuracy.

5.2 Experiment 2

Results by training the model on a dataset of two years:

Epoch Number	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.2702	0.8992	0.2541	0.9007
2	0.2518	0.9037	0.2580	0.9004
3	0.2494	0.9040	0.2497	0.9011

The subsequent table illustrates the lack of enhancement in accuracy during the second epoch, indicating the absence of overfitting in the training procedure. In comparison to the first experiment, it is evident that there is a significant reduction in loss. The results of this experiment demonstrate the ability of the model to progressively improve pattern recognition capabilities, as reflected by the notable enhancement of accuracy as the number of training epochs increases.

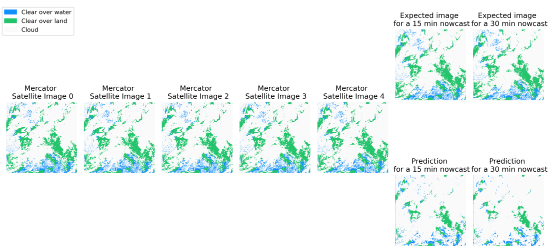


Fig. 4. Results of Experiment 6: 30 min nowcasting of Cloud Coverage by adding the 15min nowcasting result to the input. The image was produced based on a dataset of one month.

5.3 Experiment 3

Results by training the model on a dataset of one month:

Epoch Number	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.6024	0.7076	0.5825	0.7287
2	0.5903	0.7173	0.5822	0.7288

Results by training the model on a dataset of two years:

Epoch Number	Loss	Accuracy	Validation Accuracy	Validation Loss
1	0.2531	0.8973	0.2448	0.8991

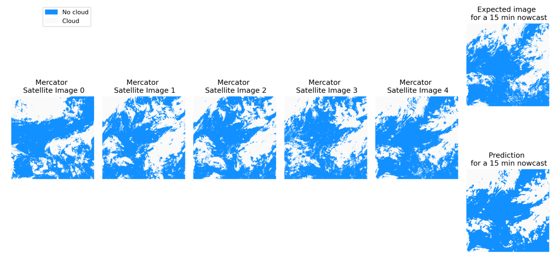


Fig. 5. Results of Experiment 3: 15 min nowcasting of Cloud Coverage by providing input and output as binary information of cloud/ no cloud. The image was produced based on a dataset of one month.

Despite the removal of information related to land and sea classes from the input provided to the model, the model performance is not significantly impaired as compared to when it was provided with the four original classes. The proportions between the other parameters: loss, accuracy, validation loss, remain minimally affected.

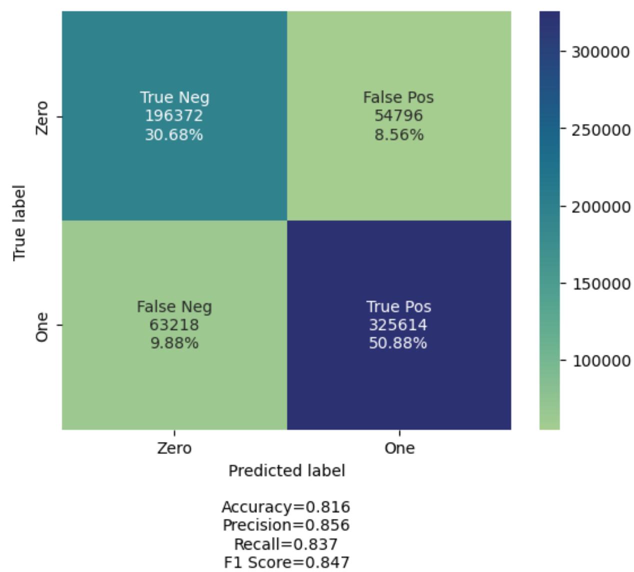


Fig. 6. Confusion Matrix for a binary output of a 15min nowcast. One stands for cloud and zero stands for no cloud. The label represents the ground truth for the cloud coverage and the predicted label represents the cloud coverage prediction by the model.

Our calculation of the Intersection over Union (IoU) score based on the confusion matrix resulted in a value of 0.734 percent. Despite obtaining an accuracy of 89.91 percent, the lower IoU score suggests that the performance of the model may be a bit lower than depicted by the overall accuracy. This is due to the fact that the IoU score provides a more nuanced evaluation of the relationship between the predicted and the ground truth pixels, whereas accuracy only provides a binary measure of correct and incorrect predictions.

5.4 Experiment 4

Results by training the model on a dataset of one month:

Epoch Number	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.7220	0.6829	0.6303	0.7258
2	0.6227	0.7232	0.6100	0.7318
3	0.6225	0.7134	0.6113	0.7235
4	0.6139	0.7247	0.6128	0.7236
5	0.6120	0.7247	0.5940	0.7322

It is noteworthy that the model has a lower accuracy when utilizing dropout. Despite this, it is apparent that the model performs robustly, achieving an accuracy of approximately 73 percent over 5 epochs, even when dropping out 20 percent of the output from the layers.

After thorough research, we have obtained results relevant to our primary research question:

(1) Our first sub research question, "How does the data produced by meteorological satellites covering Europe appear?" is depicted in Figure 4 and 5, which illustrates the results of our most successful experiments predicting cloud coverage in real time.
(2) Our second sub research question, "Which neural network architectures are used for image-to-image prediction?" is addressed in the Related Works Section, where we identify commonly used machine learning architectures in the field and demonstrate the application of ConvLSTM in our research.
(3) Our third sub research question, "How effective is ConvLSTM for cloud coverage forecasting?" is answered through the results from our experiments.

5.5 Discussion

Upon examination of the results of experiment two which yields the highest accuracy in the prediction of cloud coverage, it is evident that, when compared to the expected images, the image of the 15 minute nowcast, as well as the 30 minute nowcast shows a tendency to cover more areas of the image with clouds. This is due the fact that the cloud class is the dominant class in the ground truth.

The use of the sparse categorical cross-entropy loss function in next frame prediction task may result in a tendency for the model to falsely assign class labels to pixels in the prediction. While comparing the predicted and ground truth images it is particularly noticeable that most areas are typically covered with clouds. This is due to the function's method of comparing the predicted class probabilities to the true class label and adjusting the model's weights to minimize the difference between the two.

In future work, it may be beneficial to explore further potential solutions and to conduct additional validation analyses for resolving this issue. The first solution could involve modifying the existing loss function, which is based on intersections between clouds. This can be achieved by incorporating loss functions such as Dice Loss, Jaccard Loss, Intersection-over-Union (IoU) loss. Another approach to addressing class imbalance is to implement balanced training classes with assigned weights reflecting their frequency and the desired level of amplification.

Upon conducting a visual analysis of the second experiment, we observe that the predicted image of cloud coverage does not fully depict detailed information of land and sea, instead the model tends to display clouds. To further investigate this, we did conduct experiment three in which the model only provides information on clouds and no cloud areas. Despite this modification, the results of both experiments showed a very similar results of accuracy. This prompts us to consider alternative methods for training the model in future work. Potential solutions include incorporating additional meteorological parameters and training the model on geographic information prior to cloud mask information. Furthermore, we could consider increasing the amount of training blocks. Although having a lower amount of blocks can prevent overfitting, having a higher number of blocks can result in more comprehensive training data and more accurate predictions. To implement this approach, a sliding window approach could be used to generate the blocks, providing more combinations and variety in the training data.

6 CONCLUSION

In conclusion, our findings indicate that ConvLSTM can effectively be used for predicting cloud coverage in subsequent frames. With an overall validation accuracy of 90,11 percent, ConvLSTM has proven to be an efficient solution to next frame prediction. However, to attain both a high overall accuracy and a visually accurate depiction of details such as sea and land together with clouds that closely match the expected image, further adjustments may be necessary. These avenues for improvement can be explored in future studies.

REFERENCES

[1] [n. d.]. 9. Classes. https://docs.python.org/3/tutorial/classes.html#generators
[2] [n. d.]. datetime — Basic date and time types. https://docs.python.org/3/library/datetime.html
[3] [n. d.]. Drops. https://www.drops.live/
[4] [n. d.]. EUMETSAT Product Navigator. https://navigator.eumetsat.int/product/EO:EUM:DAT:MSG:HRSEVIRI
[5] [n. d.]. functools — Higher-order functions and operations on callable objects. https://docs.python.org/3/library/functools.html
[6] [n. d.]. Guide | TensorFlow Core. https://www.tensorflow.org/guide
[7] [n. d.]. Guide to the WMO Table Driven Code Form Used for the Representation and Exchange of Regularly Spaced Data In Binary Form: . https://old.wmo.int/extranet/pages/prog/www/WMOCodes/Guides/GRIB/GRIB2_062006.pdf
[8] [n. d.]. Installation — pygrib documentation. https://jswhit.github.io/pygrib/installing.html#dependencies
[9] [n. d.]. math — Mathematical functions. https://docs.python.org/3/library/math.html
[10] [n. d.]. Miscellaneous operating system interfaces. https://docs.python.org/3/library/os.html
[11] [n. d.]. NumPy documentation — NumPy v1.24 Manual. https://numpy.org/doc/stable/
[12] [n. d.]. Pyresample — pyresample 1.26.0.post0+21.g8393134.dirty documentation. https://pyresample.readthedocs.io/en/latest/
[13] [n. d.]. shutil — High-level file operations. https://docs.python.org/3/library/shutil.html
[14] [n. d.]. The 0 degree service is the main mission of Meteosat Second Generation. https://www.eumetsat.int/0-degree-service
[15] [n. d.]. Use a GPU | TensorFlow Core. https://www.tensorflow.org/guide/gpu
[16] [n. d.]. Users guide — Matplotlib 3.6.3 documentation. https://matplotlib.org/stable/users/index.html
[17] [n. d.]. warnings — Warning control. https://docs.python.org/3/library/warnings.html

[18] [n. d.]. zipfile — Work with ZIP archives. https://docs.python.org/3/library/zipfile.html
[19] 2009. Cloud mask product over full Earth disk. https://data.eumetsat.int/product/EO:EUM:DAT:MSG:CLM
[20] 2015. MTG-FCI: ATBD for Cloud Mask and Cloud Analysis Product. https://www-cdn.eumetsat.int/files/2020-06/pdf_atbd_cloud_mask_analysis.pdf
[21] Alexander Albrecht and Felix Naumann. [n. d.]. Managing ETL Processes. https://hpi.de/fileadmin/user_upload/fachgebiete/naumann/publications/2008/ETL_Management.pdf
[22] Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rüdiger Wirth. 1999. The CRISP-DM user guide. In *4th CRISP-DM SIG Workshop in Brussels in March*, Vol. 1999. sn.
[23] Aidan Clark, Jeff Donahue, and Karen Simonyan. 2019. Adversarial Video Generation on Complex Datasets. *arXiv: Computer Vision and Pattern Recognition* (9 2019).
[24] Casper Da Costa-Luis. [n. d.]. tqdm documentation. https://tqdm.github.io/
[25] family=Agrawal given i=S.A., " given i=L.B., "family=Barrington, " given i=C.B., "family=Bromberg, family=Burge given i=J.B., family=Gazen given i=C.G., and family=Hickey given i=J.H. [n. d.]. Machine Learning for Precipitation Nowcasting from Radar Images. https://arxiv-org.ezproxy2.utwente.nl/abs/1912.12132
[26] Seongchan Kim, Seungkyun Hong, Minsu Joh, and Sa-Kwang Song. 2017. DeepRain: ConvLSTM Network for Precipitation Prediction using Multichannel Radar Data. *arXiv: Learning* (11 2017).
[27] Dina Lazorenko. [n. d.]. Reprojection Quiz. https://ysxhk0pw57v.typeform.com/to/SHKuHoZ7
[28] J. Leinonen, U. Hamann, U. Germann, and J. R. Mecikalski. 2022. Nowcasting thunderstorm hazards using machine learning: the impact of data sources on performance. *Natural Hazards and Earth System Sciences* 22, 2 (2022), 577–597. https://doi.org/10.5194/nhess-22-577-2022
[29] J. Leinonen, U. Hamann, U. Germann, and J. R. Mecikalski. 2022. Nowcasting thunderstorm hazards using machine learning: the impact of data sources on performance. *Natural Hazards and Earth System Sciences* 22, 2 (2022), 577–597. https://doi.org/10.5194/nhess-22-577-2022
[30] Mantrir Mantrir. [n. d.]. GitHub - mantrir20/Cloud-Cover-Prediction: Next Sequence Prediction of Satellite Images using Neural Networks. https://github.com/mantrir20/Cloud-Cover-Prediction
[31] Johan Mathe, Peter Dudfield, and Jacob Bieker. 2022. Skillful Nowcasting with Deep Generative Model of Radar (DGMR). https://github.com/openclimatefix/skillful_nowcasting/tree/main/dgmr/layers
[32] Aditi Mittal. 2022. Understanding RNN and LSTM - Aditi Mittal. https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e
[33] Rajin Ramphul. 2018. Precipitation prediction using ConvLSTM deep neural network. https://medium.com/@rajin250/precipitation-prediction-using-convlstm-deep-neural-network-b9e9b617b436
[34] Suman Ravuri. 2021. Skilful precipitation nowcasting using deep generative models of radar. https://www.nature.com/articles/s41586-021-03854-z?error=cookies_not_supported&code=af4ee9c0-4c72-4c9a-8865-6c655acd2e32
[35] Suman V. Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Rémi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, Rachel Prudden, Amol Mandhane, Aidan Clark, Andrew Brock, Karen Simonyan, Raia Hadsell, Niall H. Robinson, Ellen Clancy, Alberto Arribas, and Shakir Mohamed. 2021. Skillful Precipitation Nowcasting using Deep Generative Models of Radar. *CoRR* abs/2104.00954 (2021). arXiv:2104.00954 https://arxiv.org/abs/2104.00954
[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Lecture Notes in Computer Science* (2015), 234–241. 28 https://doi.org/10.1007/978-3-319-24574-4\{_
[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Lecture Notes in Computer Science* (2015), 234–241. 28 https://doi.org/10.1007/978-3-319-24574-4\{_
[38] Xingjian Shi, 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. *Cornell University - arXiv* (12 2015). https://doi.org/10.48550/arxiv.1506.04214
[39] Keras Team. [n. d.]. Keras: the Python deep learning API. https://keras.io/
[40] The Nowcasting Team. 2021. Nowcasting the next hour of rain. https://www.deepmind.com/blog/nowcasting-the-next-hour-of-rain
[41] Adriaan van Natijne, Roderik Lindenbergh, and Thom Bogaard. 2020. Machine Learning: potential for local and regional deep-seated landslide nowcasting. *Sensors* (3 2020). https://doi.org/10.5194/egusphere-egu2020-19515
[42] Adriaan L. van Natijne, Roderik C. Lindenbergh, and Thom A. Bogaard. 2020. Machine Learning: New Potential for Local and Regional Deep-Seated Landslide Nowcasting. *Sensors* 20, 5 (2020). https://doi.org/10.3390/s20051425
[43] Alexandre Xavier. 2021. An introduction to ConvLSTM - Neuronio. https://medium.com/neuronio/an-introduction-to-convlstm-55c9025563a7
[44] Chunguang Zhang. 2022. Spatio-Temporal ConvLSTM for Crash Prediction - Towards Data Science. https://towardsdatascience.com/spatial-temporal-convlstm-for-crash-prediction-411909ed2cfa
[45] Shuyi Zhou, Wenhong Xie, Yuxiang Lu, Yuanlin Wang, Yulong Zhou, Nian Hui, and Changming Dong. 2021. ConvLSTM-Based Wave Forecasts in the South and East China Seas. *Frontiers in Marine Science* 8 (2021). https://doi.org/10.3389/fmars.2021.680079
[46] Hui-Min Zuo, Jun Qiu, Qi Wang, and Fangfang Li. 2021. Ten-Minute Prediction of Solar Irradiance Based Cloud Detection and a Long Short-Term Memory (LSTM) Model. *SSRN Electronic Journal* (2021). https://doi.org/10.2139/ssrn.3944467

A APPENDIX

A.1

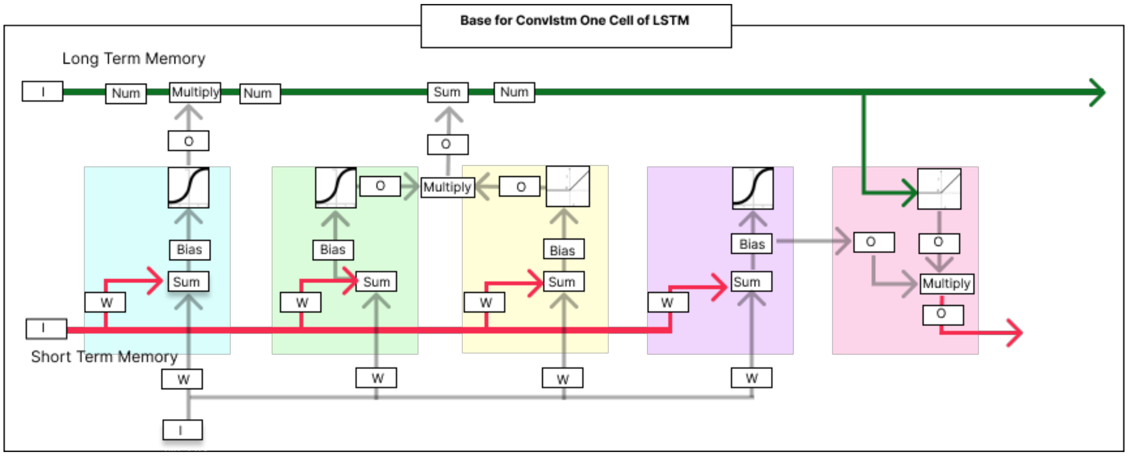


Fig. 7. Base for ConvLSTM: One Cell

B APPENDIX

B.1

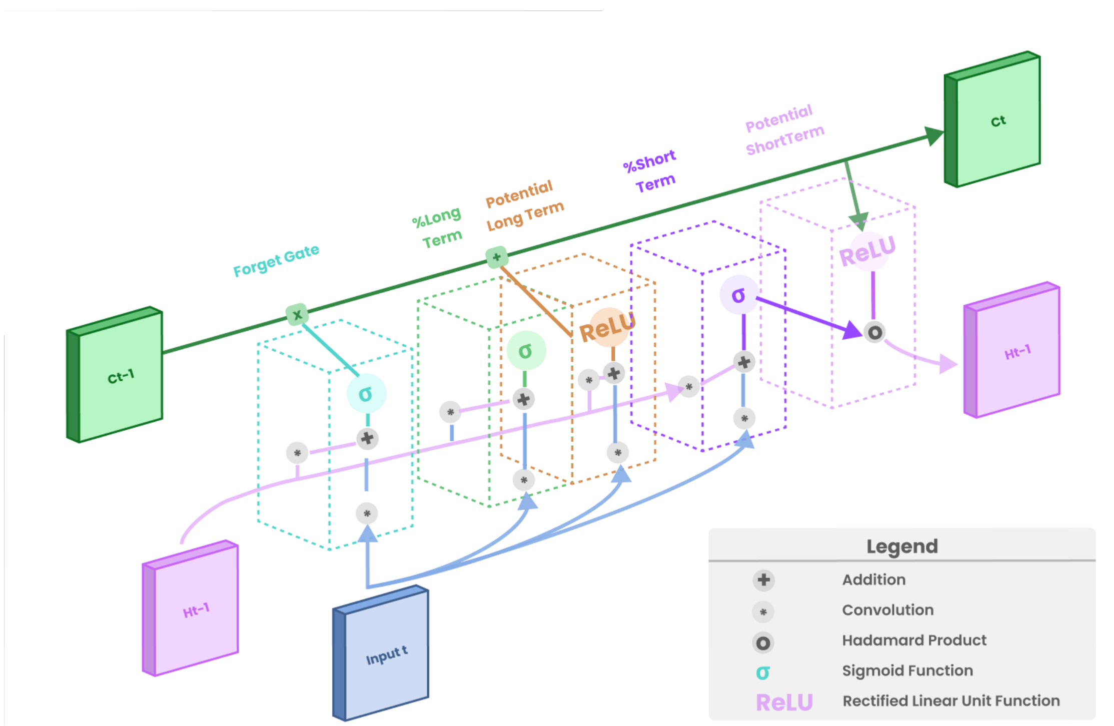
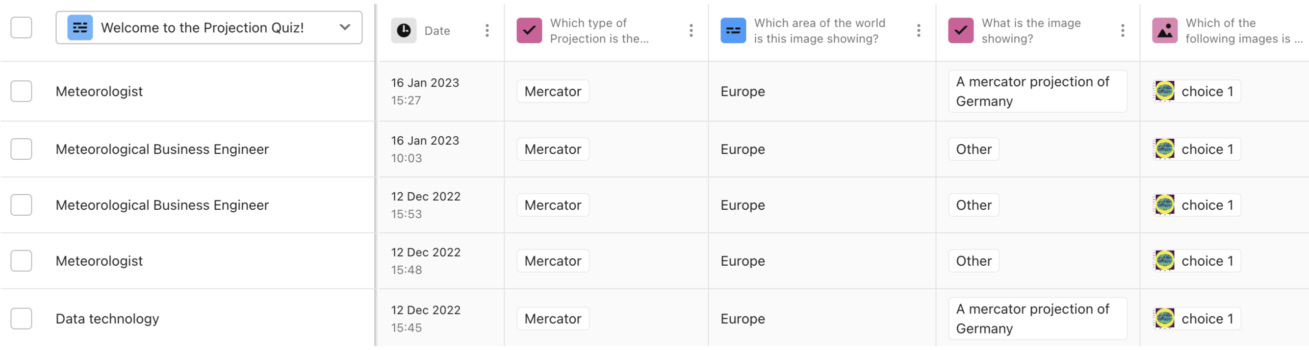


Fig. 8. ConvLSTM for our machine learning model representing the First Cell

C APPENDIX

C.1



Welcome to the Projection Quiz!	Date	Which type of Projection is the...	Which area of the world is this image showing?	What is the image showing?	Which of the following images is ...
Meteorologist	16 Jan 2023 15:27	Mercator	Europe	A mercator projection of Germany	choice 1
Meteorological Business Engineer	16 Jan 2023 10:03	Mercator	Europe	Other	choice 1
Meteorological Business Engineer	12 Dec 2022 15:53	Mercator	Europe	Other	choice 1
Meteorologist	12 Dec 2022 15:48	Mercator	Europe	Other	choice 1
Data technology	12 Dec 2022 15:45	Mercator	Europe	A mercator projection of Germany	choice 1

Fig. 9. Expert Validation Results from a customized quiz

D APPENDIX

D.1

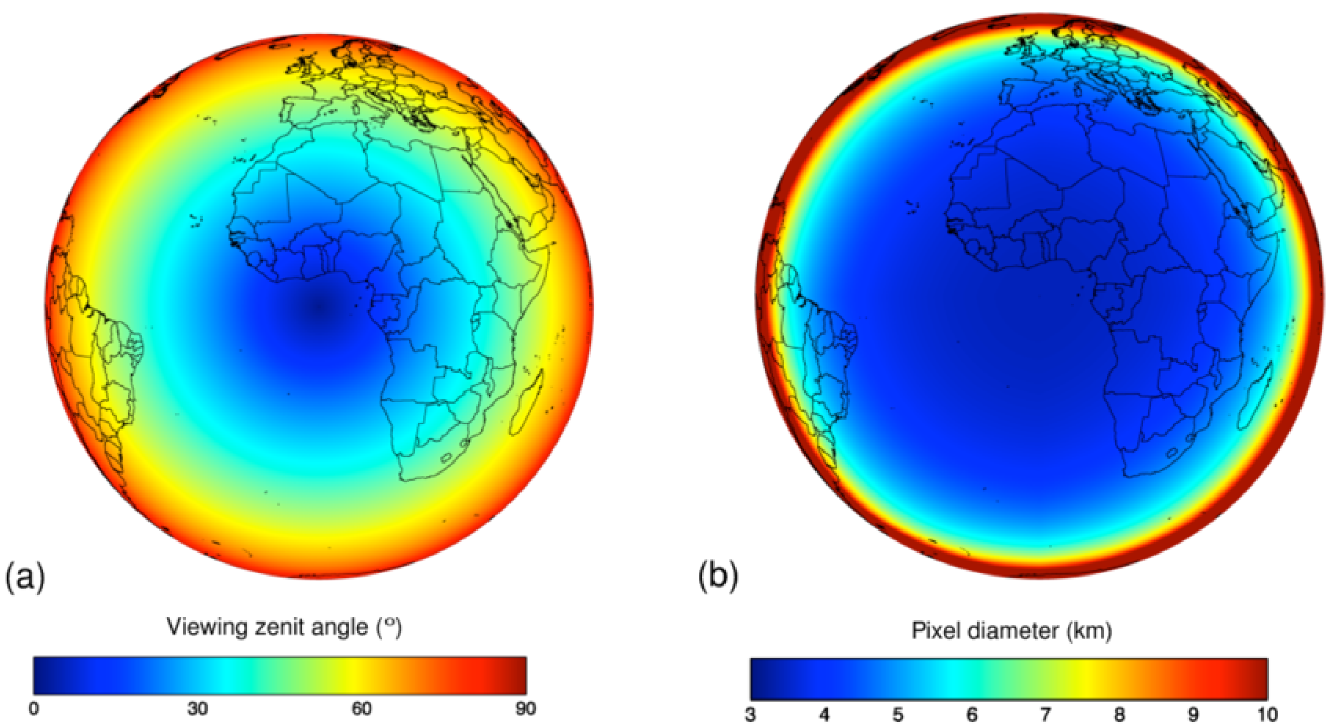


Fig. 10. Image of the Earth in geostationary projection. The image shows the amount of information of a pixel depending on its location. The blue areas shows the area of the sub-satellite point. The colors show the increasing size of area represented by a pixel in kilometeres.

E APPENDIX

Code Figure	Meaning
0	Clear over water
1	Clear over land
2	Cloud
3	No data
4-191	Reserved
192-254	Reserved for Local Use
255	Missing

Table 1. Cloud Mask Type (CODE TABLE 4.217). The Table shows the possible classes for each pixel within a satellite image.

Module:	Usage:
os	Interaction with the command line[10]
generator	To produce a sequence of data blocks for the input[1]
math	To perform mathermatical functions[9]
zipfile	To create, read, write, append, list zip files[18]
tqdm	Creating a progress bar[24]
pygrib	Reading GRIB files[8]
pyresample : image and geometry modules	Resampling and displaying geospatial image data[12]
Modules from matplotlib	Visualise data[16]
numpy	Performance of mathematical functions[11]
tensorflow	Perform tensorflow operations during training and prediction[6]
Keras: models, layers	Interface for the tensorflow library[39]
tensorflow-gpu	Running tensorflow on a gpu device[15]
shutil	Perform operations on files[13]
warnings	Applied to filter warnings[17]
datetime	Provides classes to work with date/time[2]
functools	Provider functions to work with functions[5]

Table 2. We used the following python modules for this research