

There are four main folders in the project: code, data, models and logdir.

Data

This folder contains all the data used from the two studied locations: Loc.1 (latitude=40.4°, longitude=6.0°) and Loc.2 (latitude=39.99°, longitude=-0.06°).

Sorted by year, month and day, each location has three kinds of data:

- The files named as just a number are 151x151 irradiance estimates matrices centered in the same location obtained from <http://msgcpp.knmi.nl>. The spatial resolution is 0.03° for both latitude and longitude.
- The files named Real_ are the irradiance measurements at the location
- The files named CopernicusClear_ are the clear sky estimates from the CAMS McClear model

Each file contains the 96 15-minute samples for the same day in Matlab format and UTC time.

Code

All the python scripts used to train the neural networks and perform the forecasts. The main files are:

- **tf1.yml**: List of the modules and versions used. A clean Anaconda environment created from this file can run all the code in the project.
- **learnRadiation.py**: The script to train a new model. Changing the variables “paper_model_name” and “location”. The first variable selects the kind of model to fit and the second one the training location.
- **predictOnly.py**: Loads a trained model and performs the forecast. Notice that the model and location must match the ones used to train the model stored in the “training_path” folder

Models

This folder contains all the trained models and their forecasting results. There is also a training folder to contain the last trained model.

Logdir

This folder stores Tensorboard files during training

How to train and test a model

A new model can be trained using “learnRadiation.py”. This script has three parameters

- **location:** Selects the location where the model will be trained (LOC1 or LOC2)
- **paper_model_name:** This sets the inputs to match the ones used in the models from the article.
- **training_path:** The folder to save the trained model

Then the “predictOnly.py” script allows performing the forecasts. It is important to set the same parameters as in the “learnRadiation.py” script. This program will generate the predictions and save them in the model folder. It also plots some days, which can be modified at the bottom of the script.

For instance for LOC2 and model *TOA & all real* we would run:

```
"python learnRadiation.py TOAallreal LOC2 training"
```

This will train the neural network and save the results in the folder models/training.

After this, we would generate the results and plot some days using:

```
“python predictOnly.py TOAallreal LOC2 training”
```

This will save the forecasts and real values in the training folder and show figures with 1 to 6 hour forecasts

The models used for the article can also be evaluated by using predictOnly.py and targeting their folders. For instance, to evaluate the *TOA & all real* model used in the article, this command must be used:

```
“python predictOnly.py TOAallreal LOC2 RtoaAllReal”
```