1. Usage guide for general_tamsat_alert

1.1. Installation

1.1.1. From PyPI

```
pip install general-tamsat-alert
```

1.1.2. From source

```
git clone https://github.com/brightlego/general_tamsat_alert.git
cd general_tamsat_alert
python3 -m build
python3 -m pip install dist/*.whl
```

1.2. Documentation

Requirements:

The code has been tested on Python version 3.11 and is not guaranteed to work with other python versions.

The following modules are required to install general_tamsat_alert:

- xarray
- numpy
- scipy
- fastroc

Additional requirement for setting input parameters for some functions:

- datetime

Running the code:

The code runs via the function do_forecast(), which takes in a netcdf file containing time series data and outputs an xarray object containing an ensemble forecast, and associated statistics.

Input parameters:

:param datafile: netcdf file containing the time series data on which to base the forecasts. The datafile must include a time axis, but the format is otherwise flexible

:param field_name: name of the variable to be forecast

:param init_date: initiation date of the forecast (datetime object)

:param poi_start: date of the start of the period of interest (datetime object)

:param poi_end: date of the end of the period of interest (datetime object)

:param time_label [default 'time']: time axis label in the netcdf file

:param period [default 12]: period of the data to be used for deriving the climatology

:param weights_flag [default 0]: type of ensemble weighting to be used:

- 0: No weighting
- 1: Weighting using the proximity of the ensemble member year to the initiation date
- 2: Weighting using a monthly data included in weighting_data_file

:param weighting_data_file [default 'None']: text file containing the data to be used for weighting. The data are in the format used for the NOAA composite and correlation site (format described here: https://psl.noaa.gov/data/composites/createtime.html)

:param weighting_strength [default 1]: coefficient specifying the strength of the weighting used when weights_flag is set to 1 or 2. 0 indicates no weighting; floats >0 indicates weighting is applied. Users should experiment to find the most appropriate weighting strength

:param do_increments [default 1]: flag specifying whether or not the ensemble members should be incremented from the initial state. Set do_increments to 0 for no incrementing; 1 for incrementing

Returns:

xarray dataset on the same grid and using the same dimensions as datafile, with an additional dimension 'ensemble' specifying the ensemble number. The dataset includes the following variables:

ensemble_out: array containing the full forecast ensemble (dimensions <datafile geographical dimensions>, <datafile time dimension>, ensemble)

weights: array containing the the weights applied to each ensemble member at each point in space (dimensions <datafile geographical dimensions>, ensemble). Note that in the current version of the code, weights is constant over the geographical domain

ens_mean: weighted ensemble mean (dimensions <datafile geographic dimensions>)

ens_std: weighted ensemble standard deviation (dimensions <datafile geographic dimensions>)

clim: climatology of the data in datafile (based on the user specified periodicity)

Example function call:

```
import datetime as dtmod

from general_tamsat_alert import do_forecast


field_name='precip'

time_label='time'

datafile='pr_gpcc_africa.nc'
```

```
init_date=dtmod.datetime(1997,9,1)
poi_start=dtmod.datetime(1997,10,1)
poi_end=dtmod.datetime(1997,10,1)
period=12
weights_flag=2
weighting_data_file='oni.data'
do_increments=0
weighting_strength=1

tmpout=do_forecast(datafile,field_name,init_date,poi_start,poi_end,
                time_label,period,weights_flag,weighting_data_file,
                weighting_strength,do_increments)
```

The example function call uses regridded and subset GPCC precipiation data, and the Oceanic Nino Index provided by NOAA. Convenience copies of these datasets can be found in https://gws-access.jasmin.ac.uk/public/tamsat/tamsat_alert/example_data/

1.2.1. Further details about the date inputs and the ensemble statistics:

Three dates need to be specified by the user:

- init_date is the date on which the user sets off the forecast. It is assumed that the values of variable to be forecast are unknown after init_date
- poi_start is the start of the user's period of interest (for example, the start of the growing season)
- poi_end is the end of the user's period of interest (for example, harvest date)

Note that:

1. the period of interest can be either entirely in the future (i.e. after init_date) or partially in the past and partially in the future. The system does not allow users to specify a period of interest entirely in the past. These concepts are illustrated below in Figure 1.
2. If the period of interest start and end (poi_start and poi_end) are set to the same date, a snapshot forecast is produced for a single date in the future.
3. The ensemble statistics output by do_forecast() are derived for the period of interest only - i.e. ens_mean and ens_std are the ensemble mean and standard deviation of all of the ensemble members during the period of interest.
4. The length of the forecast is determined automatically by the system as the maximum period encompassed by poi_start, init_date and poi_end. It is not possible to run the forecasts beyond the end of period of interest.

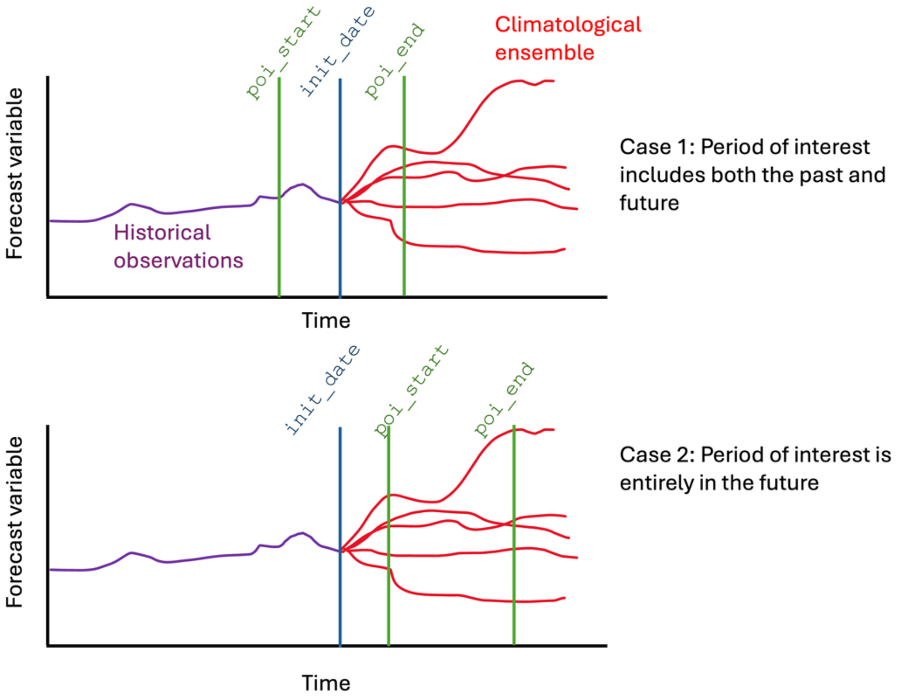

Figure 1: Sketch of TAMSAT-ALERT ensemble forecast output, illustrating the dates input by the user

1.2.2. Demo

A demonstration of the code for making SST and precipitation forecasts is available at: https://gws-access.jasmin.ac.uk/public/tamsat/tamsat_alert/gmd_paper/demo.zip The demo includes a jupyter notebook and the required netcdf data files