# Generating bespoke forecasts with the TAMSAT-ALERT Python Tool

*NOTE*

We advise that you complete the '**Reproducing Testcases.pdf'** activities before continuing with this worksheet to develop your own forecast. This is to ensure that your system is operating correctly and makes it easier for the TAMSAT team to troubleshoot any problems you may have.

**Recap:**

The TAMSAT-ALERT Python Tool is a piece of software (Python code) designed to allow users to easily generate TAMSAT-ALERT soil moisture forecasts for their period and region of interest. The software is designed so that users only need to type one line of code and the software does all the rest!

Users specify the period and region of interest, and the meteorological tercile forecast, and the Python Tool will download the required data, apply the weighting, and output your forecast in a number of formats.

The TAMSAT team intend to further develop the tool in future to improve functionality and fix any bugs, but by providing the software it this way, it will be easy for users to keep up-to-date with the latest version whilst continuing to operationally produce soil moisture forecasts.

In this activity, you will use the TAMSAT-ALERT Python tool to generate your own bespoke forecasts for a region and period of interest. The process of running the tool will be the same as when reproducing test-cases: there is no need to edit the underlying Python script and you will not need to re-install the packages, you will simply be adjusting the single line of code used to run the forecasts.

Once you are ready to get started generating your bespoke forecasts, follow the steps below.

## 1) Specifying your 'region of interest':

The TAMSAT-ALERT tool uses data developed for all of Africa. Your region of interest can therefore include any part of the continent. This could be a region (e.g. West Africa), country (e.g. Ghana) or sub-national designation (e.g. the Ashanti region of central Ghana).

When choosing your region, bear in mind the larger the region, the longer it will take for the code to run and the bigger the outputs data files (.nc) will be. We recommend starting with a country or sub-national designation.

Once you have decided on your region of interest, you need to identify the bounding longitudes and latitudes. You will need the minimum longitude, maximum longitude, minimum latitude and maximum latitude of the region. If you don't know what these coordinates are, you can use Google Maps to locate the region of interest. Right click on the map and choose 'What's here?' to display the longitude and latitude of a point.

If your longitudes are west of the Prime Meridian (0° longitude) or your latitudes are south of the Equator (0° latitude) you will need to include a minus sign (-) before the number.

You now have the **lon_min**, **lon_max**, **lat_min** and **lat_max** variables for your Python call (see below).

NOTE: We intend to build in functionality to run forecasts for a point (rather than region). This is not yet fully tested, but will be included in the next version of the tool.

## 2) Specifying a 'period of interest':

A period of interest is often a season or month of importance to agricultural production, typically a rainy season. To specify the period of interest, you will need both a start and end date. For example, if your period of interest is Kenya's 2021 March-April-May long rains, your period of interest start date would be 1st March 2021 and you end date would be 31st May 2021.

Depending on your region of interest, you should now choose a period of interest and identify the start and end dates. Bear in mind that the TAMSAT-ALERT forecasts run for 150 days. This means that your period of interest cannot extend beyond 150 days from the forecast date (see below). If it does, you will get an error message and the Python tool will not run.

You should format your period of interest start and end dates as follows:

| **Date** | **Format (YYYYMMDD)** |
|---|---|
| e.g. 23rd April 2021 | 20210423 |
| e.g. 12th January 2021 | 20210121 |

You now have the **poi_start** and **poi_end** variables for your Python call (see below).

## 3) Specifying a forecast date:

The default forecast date for operational production of TAMSAT-ALERT forecasts should always be today's date. TAMSAT-ALERT forecast files are not issued everyday (more like every 5 days), but the Python tool will find the most recent forecast file and run your forecast from this date. Your output files will be labelled with the date that the forecast file was issued, not the forecast date you specified (see below for file naming).

You should format your forecast date in the same way as your period of interest start and end dates (YYYYMMDD).

You now have the **forecast_date** variable for your Python call (see below).

## 4) Incorporating meteorological forecast information:

The TAMSAT-ALERT tool uses a tercile rainfall forecasts to weight the soil moisture ensemble forecast. For full details, refer back to plenary 3.

You can obtain a tercile rainfall forecast from any source that is relevant to your region and period of interest. For example, if interested in Kenya's 2021 March-April-May rainy season you could use ICPAC's GHACOF rainfall forecasts, Kenya Meteorological Department's seasonal rainfall outlook or another relevant product.

Weights should be proportional and should add up to 1.0. If the tercile rainfall forecast you are using instead adds up to 100, simply dived each value by 100.

You now have the **weight_up**, **weight_mid** and **weight_low** variables for your Python call (see below).

If you cannot find a relevant tercile rainfall forecast or do not want to weight your soil moisture ensemble, then you should use an even weighting of 0.33, 0.34, 0.33 for weight_up, weight_mid and weight_low, respectively.

## 5) Inputting variables into the Python call:

You now have all the information you require to run your own bespoke TAMASAT-ALERT forecast. Below is a reminder of the format of the Python call.

To run the TAMSAT-ALERT Python Tool you need only type a single line of code into the 'Anaconda Prompt' window. The format of that line of code is as follows:

```
python filepath forecast_date poi_start poi_end weight_up weight_mid, weight_low roi lon_min lon_max lat_min lat_max
```

| Argument | Description | Format | Example |
|---|---|---|---|
| **filepath** | The file path where the TAMSAT-ALERT Python Tool is stored and the file name of the TAMSAT-ALERT Python Tool. The file name will always be the same ("T-A_API.py") | String | **F:/ TAMSAT-ALERT_Python_Tool/T-A_API.py** |
| **forecast_date** | The date of the forecast. This can be any date between 1st January 2021 and today's date. | YYYYMMDD | **20210120**<br>(e.g. 20th January 2021) |
| **poi_start** | The start date of the period of interest. | YYYYMMDD | **20210301**<br>(e.g. 1st March 2021) |
| **poi_end** | The end date of the period of interest. | YYYYMMDD | **20210531**<br>(e.g. 31st May 2021) |
| **weight_up*** | The probability of upper tercile rainfall for the period of interest. This can be obtained, for instance, from the GHACOF rainfall forecast. | Numeric float | **0.33** |
| **weigth_mid*** | The probability of mid tercile rainfall for the period of interest. | Numeric float | **0.34** |
| **weight_low*** | The probability of lower tercile rainfall for the period of interest. | Numeric float | **0.33** |
| **roi** | Region of interest. A 'region' (i.e. bounding box) or 'point'. | String | **region** or **point** |
| **lon_min** | If roi = 'region', this is the minimum longitude of the bounding box. If roi = 'point', this is the longitude of that point. | Numeric float | **33.5** |
| **lon_max** | If roi = 'region', this is the maximum longitude of the bounding box. If roi = 'point', this should be 'NA'. | Numeric float | **42.0** or **NA** |
| **lat_min** | If roi = 'region', this is the minimum latitude of the bounding box. If roi = 'point', this is the latitude of that point. | Numeric float | **-4.8** |
| **lat_max** | If roi = 'region', this is the maximum latitude of the bounding box. If roi = 'point', this should be 'NA'. | Numeric float | **5.5** or **NA** |

* Weightings should add up to 1.00

Using your information for the region and period of interest, the forecast date and meteorological forecast, type the relevant Python call into the 'Anaconda Prompt' window. For example:

```
python F:/TAMSAT-ALERT_Python_Tool/T-A_API.py 20210120 20210301 20210531 0.33 0.34 0.33 region 33.5 42.0 -4.8 5.5
```

This will take a short while to run (10-20 minutes depending on the size of the region you have chosen) and once finished, you should see some new files created in the 'outputs' folder.

**Output file naming conventions:** <filename>_<poi>_<forecastdate>_<region>.<filetype>

The <filename> and <filetype> labels give the name and type of the output file. See the table below for a description of each file.

The <poi> labels refers to the period of interest (poi_start, poi_end) you specified in the Python call. This takes the format of <poi_months><poi_year>. For Kenya's 2021 March-April-May season, the <poi> label will therefore be 'MAM2021' (the first letter of each month is given to indicate the season).

The <forecastdate> label refers to the forecast date from which the forecast was run in the YYYYMMDD format. This may be different from the forecast_date you specified in the Python call as the TAMSAT-ALERT tool will find the most recent forecast file and reset the forecast_date to the date it was issued.

The <region> label gives the bounding coordinates of you region of interest, as specified in your Python call.

| **File** | **Description** |
|---|---|
| clim_mean_wrsi_<poi>_<forecastdate>_<region>.nc | NetCDF file. The climatological mean soil moisture in the period of interest for each grid cell in the region of interest. |
| clim_sd_wrsi_<poi>_<forecastdate>_<region>.nc | NetCDF file. The climatological standard deviation of soil moisture in the period of interest for each grid cell in the region of interest. |
| ens_mean_wrsi_<poi>_<forecastdate>_<region>.nc | NetCDF file. The forecast ensemble mean soil moisture for the forecast period for each grid cell in the region of interest. |
| ens_sd_wrsi_<poi>_<forecastdate>_<region>.nc | NetCDF file. The forecast ensemble standard deviation of soil moisture for the forecast period for each grid cell in the region of interest. |
| ensemble_forecast_wrsi_<poi>_<forecastdate>_<region>.nc | NetCDF file. The forecasted soil moisture each day throughout the forecast period for each ensemble member (15). |
| prob_lower_tercile_<poi>_<forecastdate>_<region>.nc | NetCDF file. The probability that soil moisture in the forecast period with fall into the lowest tercile, for each grid cell in the region of interest |
| map_plot <poi>_<forecastdate>_<region>.png | Image file. A map of the region of interest showing the soil moisture climatology, soil moisture forecast and the forecast anomaly (presented as a percentage of the long-term mean). |
| prob_map_plot <poi>_<forecastdate>_<region>.png | Image file. A map of the region of interest showing the probability that soil moisture in the forecast period will fall into the lowest tercile. |
| probdist_<poi>_<forecastdate>_<region>.png | Image file. The probability distribution of forecasted soil moisture compared to climatology. |
| timeseries_ <poi>_<forecastdate>_<region>.png | Image file. A time series of ensemble soil moisture forecast compared to the climatology. |
| terciles_ <poi>_<forecastdate>_<region>.txt | Text file. The probabilities of soil moisture in the forecast period falling into either the lower, mid or upper tercile. |

You have successfully generated your own TAMSAT-ALERT forecasts!

The output files are available for you to use as you wish. The netCDF (.nc) files contain the forecast data. You may wish to use the files to generate your own plots or figures.

The database of historic and forecast data is being regularly updated. Therefore, if you wish to begin producing TAMSAT-ALERT forecasts operationally for your region of interest, you can run the TAMSAT-ALERT Python Tool at regular intervals (e.g. weekly or monthly) to receive an updated soil moisture forecast.

The TAMSAT team will continue to be on-hand to answer any emerging questions you may have or to troubleshoot any problems. Contact us at the email addresses below. We also love to hear your experiences with using TAMSAT-ALERT data in your own activities – keep in touch!

**Vicky Boult:** v.l.boult@reading.ac.uk

**Emily Black:** e.c.l.black@reading.ac.uk

**Ross Maidment:** r.i.maidment@reading.ac.uk