Reproducing 'test cases' with the TAMSAT-ALERT Python Tool

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 reproduce two forecast 'test cases'. The aim is to get the Python Tool working correctly on your computer before we move on to generate bespoke forecasts.

Before you begin this activity, you will need the following:

- We have written this tutorial based on having installed Anaconda 3, an open-source Python distribution platform. Anaconda can be downloaded here: https://www.anaconda.com/products/distribution
- Storage space for around 18GB of data.
- Download the 'TAMSAT-ALERT_Python_Tool' zip folder from the GitHub repository. Unzip it and save the entire folder in a suitable location. (If you are using an external hard drive to save the data, you should save the folder to the external hard drive). Do not move, edit or delete any of the files or folders.

Installing the required Python packages:

The TAMSAT-ALERT Python Tool uses a number of Python packages. These add functionality to your Python environment. To ensure you have all the required packages installed:

1) Open 'Anaconda Prompt'. This can be found in the 'Anaconda' folder in your Start menu.
2) Type each of the following commands in turn into the Anaconda Prompt window. Some of these may already be installed or may need updating, and some will take a little while to run.

```
conda install xarray
conda install numpy
conda install matplotlib
conda install scipy
conda install pandas
conda install cartopy
conda install requests
conda install bs4
```

Once you have successfully installed all the required packages, you are ready to try reproducing the 'test cases'.

You will interact with the TAMSAT-ALERT Python Tool through 'Anaconda Prompt'. At no point will you be required to directly edit the code.

To run the TAMSAT-ALERT Python Tool and reproduce the 'test cases', 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 16th August 2022 and today's date.	YYYYMMDD	**20220921** (e.g. 21st September 2022)
poi_start	The start date of the period of interest.	YYYYMMDD	**20221001** (e.g. 1st October 2022)
poi_end	The end date of the period of interest.	YYYYMMDD	**20221231** (e.g. 31st December 2022)
weight_up*	The probability of upper tercile rainfall for the period of interest. This can be obtained, for instance, from the GHACOF seasonal rainfall forecast.	Numeric float	**0.15**
weigth_mid*	The probability of mid tercile rainfall for the period of interest.	Numeric float	**0.30**
weight_low*	The probability of lower tercile rainfall for the period of interest.	Numeric float	**0.55**
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, ✝ Point functionality not yet fully tested

Test case 1: Soil moisture forecast issued 21st September 2022 for Kenya's OND rainy season.

3) Type the following code into the 'Anaconda Prompt' window, remembering to insert your own filepath:

python <your filepath> 20220921 20221001 20221231 0.15 0.30 0.55 region 33.5 42.0 -4.8 5.5

4) This will take some time to run (several hours depending on your internet speed) as it will have to download historic and forecast data.
5) Once completed, you should find the following files in the 'outputs' folder:

File	Description
clim_mean_wrsi_OND2022_20220921_ 33.5_42.0_-4.8_5.5.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_OND2022_20220921_ 33.5_42.0_-4.8_5.5.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_OND2022_20220921_ 33.5_42.0_-4.8_5.5.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_OND2022_20220921_ 33.5_42.0_-4.8_5.5.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_OND2022_20220921_ 33.5_42.0_-4.8_5.5.nc	NetCDF file. The forecasted soil moisture each day throughout the forecast period for each ensemble member (15).
prob_lower_tercile_ OND2022_20220921_ 33.5_42.0_-4.8_5.5.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 OND2022_20220921_ 33.5_42.0_-4.8_5.5.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 OND2022_20220921_ 33.5_42.0_-4.8_5.5.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_ OND2022_20220921_ 33.5_42.0_-4.8_5.5.png	Image file. The probability distribution of forecasted soil moisture compared to climatology.
timeseries_ OND2022_20220921_ 33.5_42.0_-4.8_5.5.png	Image file. A time series of ensemble soil moisture forecast compared to the climatology.
terciles_ OND2022_20220921_ 33.5_42.0_-4.8_5.5.txt	Text file. The probabilities of soil moisture in the forecast period falling into either the lower, mid or upper tercile.

6) You can compare your plots against those stored in the 'test_cases' folder. They should be the same. If they are, well done, you have successfully reproduced test case 1. You can now move onto test case 2. If not, check that your input code matches that above. If you get any errors, please copy your input code and the error message and send it to Vicky at v.l.boult@reading.ac.uk.

Test case 2: Soil moisture forecast issued 5th October 2022 for Kenya's OND rainy season.

7) Type the required code into the 'Anaconda Prompt' window – this time, you'll have to work out what the code is for yourself. Hint: you'll only need to change the forecast date.
8) Once completed, compare your plots to the those in the 'test_cases' folder.

Once you have successfully reproduced the test cases, you are ready to develop your own bespoke forecasts. Follow instructions in 'Bespoke_Forecasting.pdf'.