iCASP

Integrated Catchment Solutions Programme

User Guide for MATLAB Codes Developed for Living with Water Partnership Catchment Telemetry Integration Project

v.1

May 2021

Authors:

Ehsan Kazemi

Vanessa Speight

Virginia Stovin

University of Sheffield

Table of Contents

1. Introduction

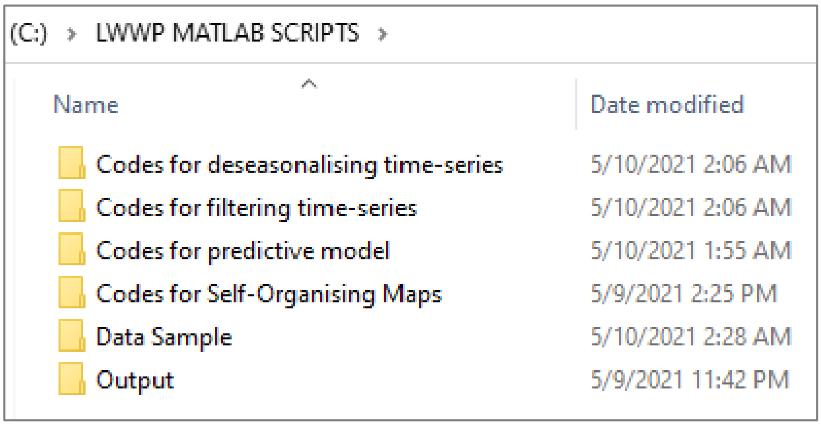
This document provides a user guide for the models developed in MATLAB 2019b for the Living with Water Partnership Catchment Telemetry Integration project. In the following, firstly, the folders where data and scripts are located are introduced in Sections 2 to 5, and then the definition of input parameters of the models and how to execute them, is presented in Sections 6 to 9, with relevant examples. For technical details and more results of the models refer to LWWP Project Final Report[1].

2. Models

There are four models presented in this document:

- Model for cleansing and filtering data (mainly water level time-series)
- Model for deseasonalising data (mainly water level time-series)
- Model for relationship exploration using Self-Organising Map (SOM)
- Model for water level predictions using Machine Learning, namely Predictive Model

The MATLAB codes for these models as well as example data and outputs are located in a folder, named *LWWP MATLAB SCRIPTS*. There are six subfolders in this folder, as shown in Figure 1, containing the scripts of the models in the first four folders, five sample data sets in folder *Data Sample*, and a folder for storing outputs of SOM and Predictive Models. For the analyses in the LWWP project, the codes of the models were executed in the order listed above, i.e. first raw data was filtered, then it was deseasonalised if required, and finally was analysed using SOM and Predictive Models (firstly with SOM for exploration of relationships within data, and then with Predictive Model for water level predictions).



(C:) › LWWP MATLAB SCRIPTS ›

Name	Date modified
Codes for deseasonalising time-series	5/10/2021 2:06 AM
Codes for filtering time-series	5/10/2021 2:06 AM
Codes for predictive model	5/10/2021 1:55 AM
Codes for Self-Organising Maps	5/9/2021 2:25 PM
Data Sample	5/10/2021 2:28 AM
Output	5/9/2021 11:42 PM

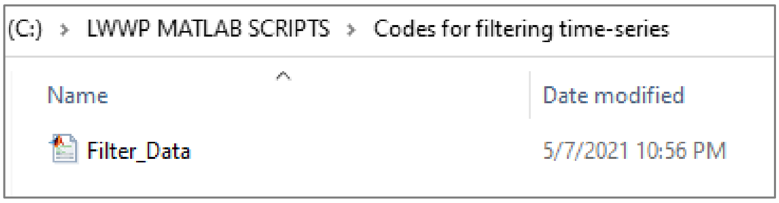
Figure 1

[1] Reference LWWP Project Final Report

3. Code Folders

The codes for filtering data, deseasonalising data, SOM analysis, and water level predictions are located in folders in *Codes for filtering time-series*, *Codes for deseasonalising time-series*, *Codes for Self-Organising Maps* and *Codes for predictive model*, respectively.

In the folder *Codes for filtering time-series*, there is one script Filter_Data.m, as shown in Figure 2, for filtering raw data.

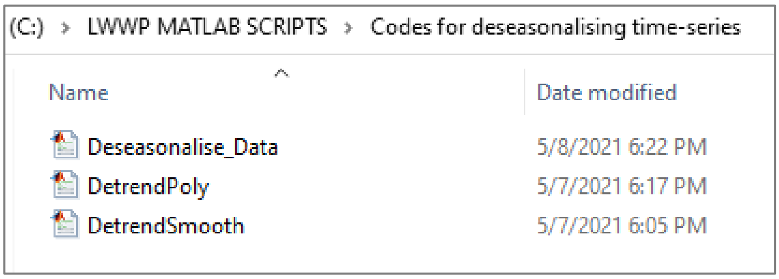


(C:) › LWWP MATLAB SCRIPTS › Codes for filtering time-series

Name	Date modified
Filter_Data	5/7/2021 10:56 PM

Figure 2

In the folder *Codes for deseasonalising time-series*, there are three scripts, as shown in Figure 3, and the main one which should be executed is Deseasonalise_Data.m.

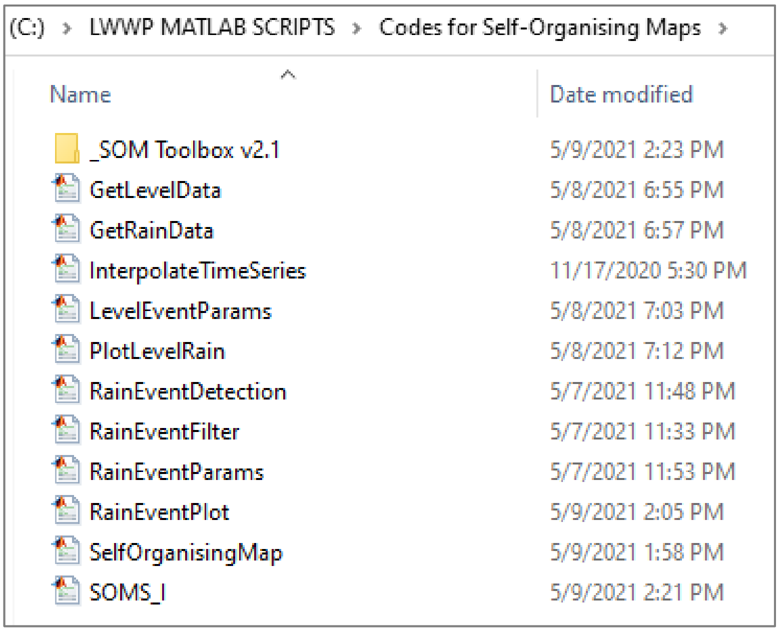


(C:) › LWWP MATLAB SCRIPTS › Codes for deseasonalising time-series

Name	Date modified
Deseasonalise_Data	5/8/2021 6:22 PM
DetrendPoly	5/7/2021 6:17 PM
DetrendSmooth	5/7/2021 6:05 PM

Figure 3

In the folder *Codes for Self-Organising Maps* there are 12 scripts, as shown in Figure 4, and the main one which should be executed is SOMS_I.m. Note that there is a folder named '*_SOM Toolbox v2.1*' here which contains the toolbox for the SOM model. This folder and all its subfolders should be added to the MATLAB path (using *Set Path* button under tab *Home* in MATLAB) to be able to use the SOM model.

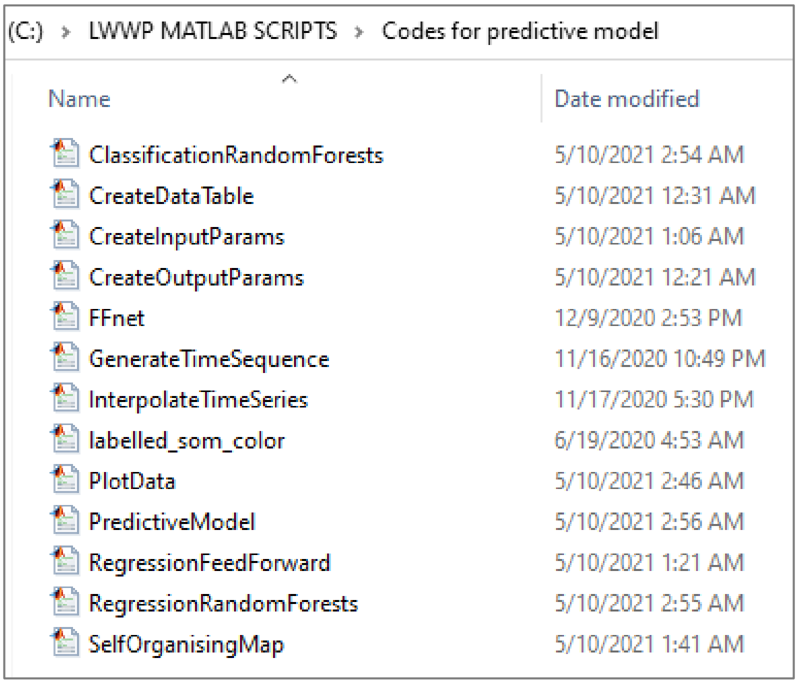


(C:) > LWWP MATLAB SCRIPTS > Codes for Self-Organising Maps >

Name	Date modified
_SOM Toolbox v2.1	5/9/2021 2:23 PM
GetLevelData	5/8/2021 6:55 PM
GetRainData	5/8/2021 6:57 PM
InterpolateTimeSeries	11/17/2020 5:30 PM
LevelEventParams	5/8/2021 7:03 PM
PlotLevelRain	5/8/2021 7:12 PM
RainEventDetection	5/7/2021 11:48 PM
RainEventFilter	5/7/2021 11:33 PM
RainEventParams	5/7/2021 11:53 PM
RainEventPlot	5/9/2021 2:05 PM
SelfOrganisingMap	5/9/2021 1:58 PM
SOMS_I	5/9/2021 2:21 PM

Figure 4

In folder *Codes for predictive model*, there are 13 scripts, as shown in Figure 5, and the main one which should be executed is PredictiveModel.m.



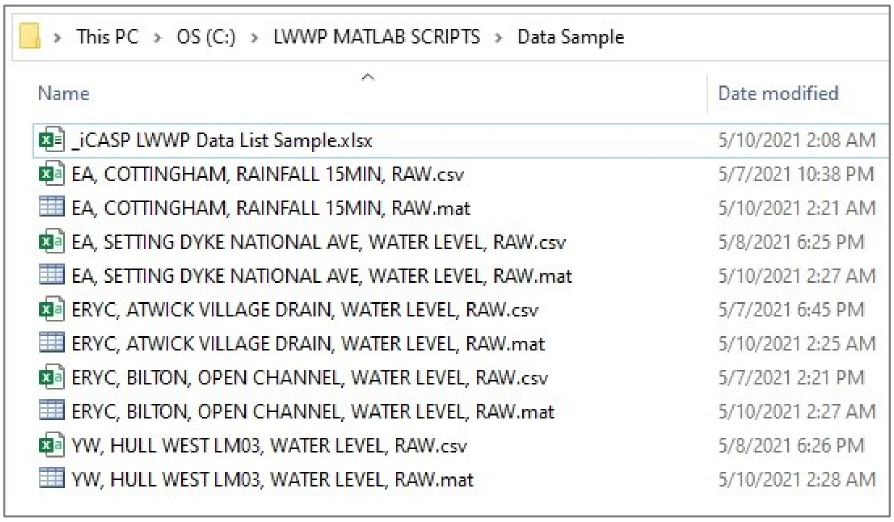
(C:) > LWWP MATLAB SCRIPTS > Codes for predictive model

Name	Date modified
ClassificationRandomForests	5/10/2021 2:54 AM
CreateDataTable	5/10/2021 12:31 AM
CreateInputParams	5/10/2021 1:06 AM
CreateOutputParams	5/10/2021 12:21 AM
FFnet	12/9/2020 2:53 PM
GenerateTimeSequence	11/16/2020 10:49 PM
InterpolateTimeSeries	11/17/2020 5:30 PM
labelled_som_color	6/19/2020 4:53 AM
PlotData	5/10/2021 2:46 AM
PredictiveModel	5/10/2021 2:56 AM
RegressionFeedForward	5/10/2021 1:21 AM
RegressionRandomForests	5/10/2021 2:55 AM
SelfOrganisingMap	5/10/2021 1:41 AM

Figure 5

4. Data Sample

Five data sets are provided to demonstrate the execution of the MATLAB codes. These data sets are located (must be located) in folder *Data Sample* as shown in Figure 6. File name of these data sets must end with ‘RAW’, and raw data must be available in both .csv and .mat formats, as some of the codes read raw data files only in CSV format and some others read them only in mat-file format.

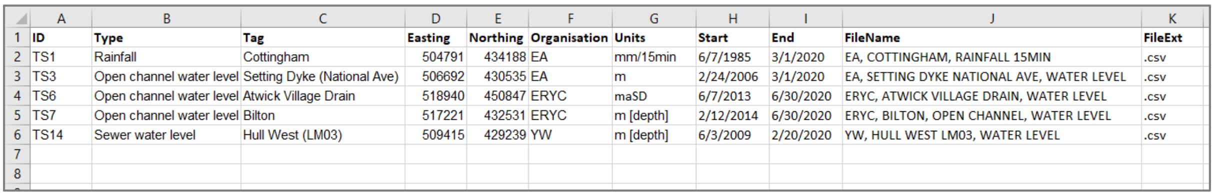


This PC > OS (C:) > LWWP MATLAB SCRIPTS > Data Sample

Name	Date modified
_iCASP LWWP Data List Sample.xlsx	5/10/2021 2:08 AM
EA, COTTINGHAM, RAINFALL 15MIN, RAW.csv	5/7/2021 10:38 PM
EA, COTTINGHAM, RAINFALL 15MIN, RAW.mat	5/10/2021 2:21 AM
EA, SETTING DYKE NATIONAL AVE, WATER LEVEL, RAW.csv	5/8/2021 6:25 PM
EA, SETTING DYKE NATIONAL AVE, WATER LEVEL, RAW.mat	5/10/2021 2:27 AM
ERYC, ATWICK VILLAGE DRAIN, WATER LEVEL, RAW.csv	5/7/2021 6:45 PM
ERYC, ATWICK VILLAGE DRAIN, WATER LEVEL, RAW.mat	5/10/2021 2:25 AM
ERYC, BILTON, OPEN CHANNEL, WATER LEVEL, RAW.csv	5/7/2021 2:21 PM
ERYC, BILTON, OPEN CHANNEL, WATER LEVEL, RAW.mat	5/10/2021 2:27 AM
YW, HULL WEST LM03, WATER LEVEL, RAW.csv	5/8/2021 6:26 PM
YW, HULL WEST LM03, WATER LEVEL, RAW.mat	5/10/2021 2:28 AM

Figure 6

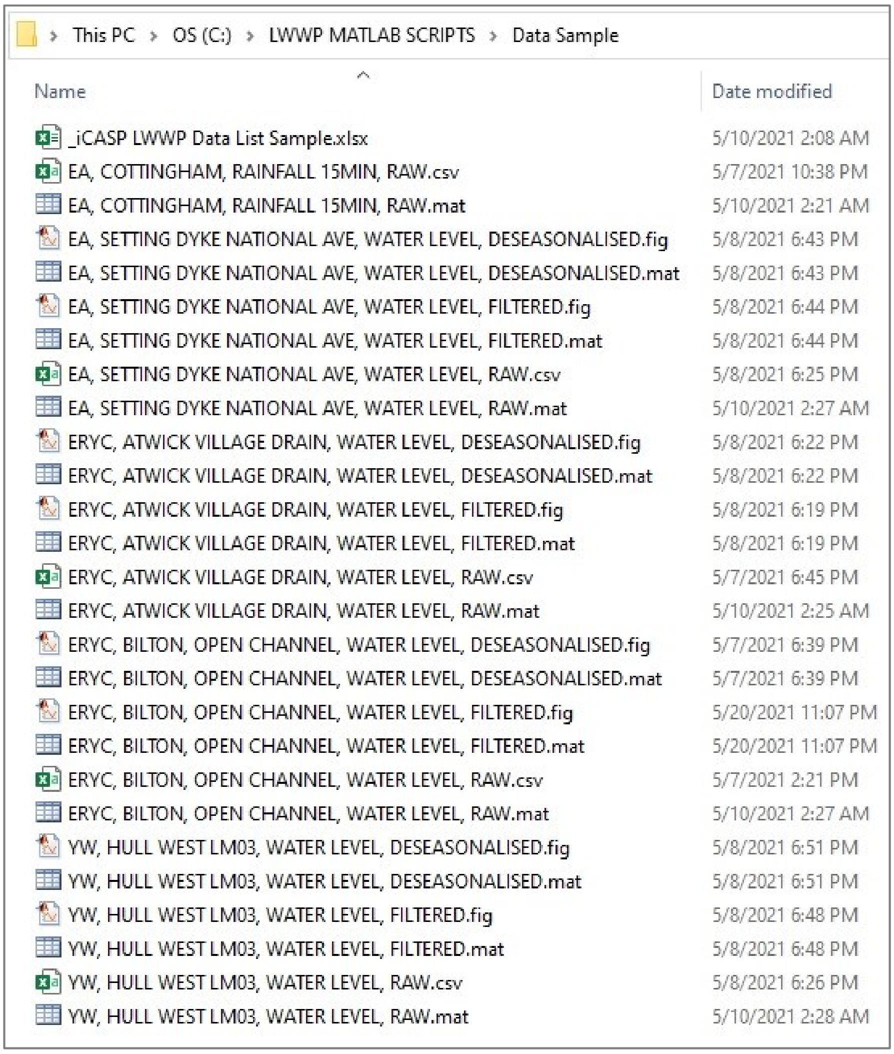
There is an Excel spreadsheet in the folder, named ‘_iCASP LWWP Data List Sample.xlsx’. This spreadsheet contains the information of the datasets: ID, tag, easting and northing of the sampling site, organisation, units of variable, start and end time of time-series, filename and extension of the raw data file initially placed in the folder, as shown in Figure 7. Note that filename should be the general name of data, not including ‘RAW’ in the end, because these filenames are called in various MATLAB scripts for all types of data, as will be shown later in this document. Raw data files (.csv and .mat) consist of two columns: a column for time stamp, and a column for value of the measured variable (water level, rainfall, etc.).



ID	Type	Tag	Easting	Northing	Organisation	Units	Start	End	FileName	FileExt
TS1	Rainfall	Cottingham	504791	434188	EA	mm/15min	6/7/1985	3/1/2020	EA, COTTINGHAM, RAINFALL 15MIN	.csv
TS3	Open channel water level	Setting Dyke (National Ave)	506692	430535	EA	m	2/24/2006	3/1/2020	EA, SETTING DYKE NATIONAL AVE, WATER LEVEL	.csv
TS6	Open channel water level	Atwick Village Drain	518940	450847	ERYC	maSD	6/7/2013	6/30/2020	ERYC, ATWICK VILLAGE DRAIN, WATER LEVEL	.csv
TS7	Open channel water level	Bilton	517221	432531	ERYC	m [depth]	2/12/2014	6/30/2020	ERYC, BILTON, OPEN CHANNEL, WATER LEVEL	.csv
TS14	Sewer water level	Hull West (LM03)	509415	429239	YW	m [depth]	6/3/2009	2/20/2020	YW, HULL WEST LM03, WATER LEVEL	.csv

Figure 7

Note that when the codes for filtering and deseasonalising data are executed (as described in Sections 6 and 0), additional files with filenames ending ‘FILTERED’ and ‘DESEASONALISED’ will be saved in the folder, as shown in Figure 8.



This PC > OS (C:) > LWWP MATLAB SCRIPTS > Data Sample

Name	Date modified
_iCASP LWWP Data List Sample.xlsx	5/10/2021 2:08 AM
EA, COTTINGHAM, RAINFALL 15MIN, RAW.csv	5/7/2021 10:38 PM
EA, COTTINGHAM, RAINFALL 15MIN, RAW.mat	5/10/2021 2:21 AM
EA, SETTING DYKE NATIONAL AVE, WATER LEVEL, DESEASONALISED.fig	5/8/2021 6:43 PM
EA, SETTING DYKE NATIONAL AVE, WATER LEVEL, DESEASONALISED.mat	5/8/2021 6:43 PM
EA, SETTING DYKE NATIONAL AVE, WATER LEVEL, FILTERED.fig	5/8/2021 6:44 PM
EA, SETTING DYKE NATIONAL AVE, WATER LEVEL, FILTERED.mat	5/8/2021 6:44 PM
EA, SETTING DYKE NATIONAL AVE, WATER LEVEL, RAW.csv	5/8/2021 6:25 PM
EA, SETTING DYKE NATIONAL AVE, WATER LEVEL, RAW.mat	5/10/2021 2:27 AM
ERYC, ATWICK VILLAGE DRAIN, WATER LEVEL, DESEASONALISED.fig	5/8/2021 6:22 PM
ERYC, ATWICK VILLAGE DRAIN, WATER LEVEL, DESEASONALISED.mat	5/8/2021 6:22 PM
ERYC, ATWICK VILLAGE DRAIN, WATER LEVEL, FILTERED.fig	5/8/2021 6:19 PM
ERYC, ATWICK VILLAGE DRAIN, WATER LEVEL, FILTERED.mat	5/8/2021 6:19 PM
ERYC, ATWICK VILLAGE DRAIN, WATER LEVEL, RAW.csv	5/7/2021 6:45 PM
ERYC, ATWICK VILLAGE DRAIN, WATER LEVEL, RAW.mat	5/10/2021 2:25 AM
ERYC, BILTON, OPEN CHANNEL, WATER LEVEL, DESEASONALISED.fig	5/7/2021 6:39 PM
ERYC, BILTON, OPEN CHANNEL, WATER LEVEL, DESEASONALISED.mat	5/7/2021 6:39 PM
ERYC, BILTON, OPEN CHANNEL, WATER LEVEL, FILTERED.fig	5/20/2021 11:07 PM
ERYC, BILTON, OPEN CHANNEL, WATER LEVEL, FILTERED.mat	5/20/2021 11:07 PM
ERYC, BILTON, OPEN CHANNEL, WATER LEVEL, RAW.csv	5/7/2021 2:21 PM
ERYC, BILTON, OPEN CHANNEL, WATER LEVEL, RAW.mat	5/10/2021 2:27 AM
YW, HULL WEST LM03, WATER LEVEL, DESEASONALISED.fig	5/8/2021 6:51 PM
YW, HULL WEST LM03, WATER LEVEL, DESEASONALISED.mat	5/8/2021 6:51 PM
YW, HULL WEST LM03, WATER LEVEL, FILTERED.fig	5/8/2021 6:48 PM
YW, HULL WEST LM03, WATER LEVEL, FILTERED.mat	5/8/2021 6:48 PM
YW, HULL WEST LM03, WATER LEVEL, RAW.csv	5/8/2021 6:26 PM
YW, HULL WEST LM03, WATER LEVEL, RAW.mat	5/10/2021 2:28 AM

Figure 8

5. Output Folder

The output of the SOM and Predictive Models are saved in folder *Output*. In this folder, there are two subfolders, namely, *SOM_I* and *Predictive*. When the codes of SOM and Predictive Model are executed, as shown in Sections 8 and 9, for each test a subfolder is created in these folders with a name which is specified in the beginning of the codes. The output files, including mat-files, figures, etc., are saved in the test folder. In addition, m-files of the executed codes are also saved in the test folder. Figure 9 shows the process for the output of the SOM model, and Figure 10 shows it for the output of the Predictive Model. In these examples, the test name is set to 'Test 01' in the codes. If a new code execution is performed with the same name, the test folder and containing files will be overwritten. For more details see Sections 8 and 9.

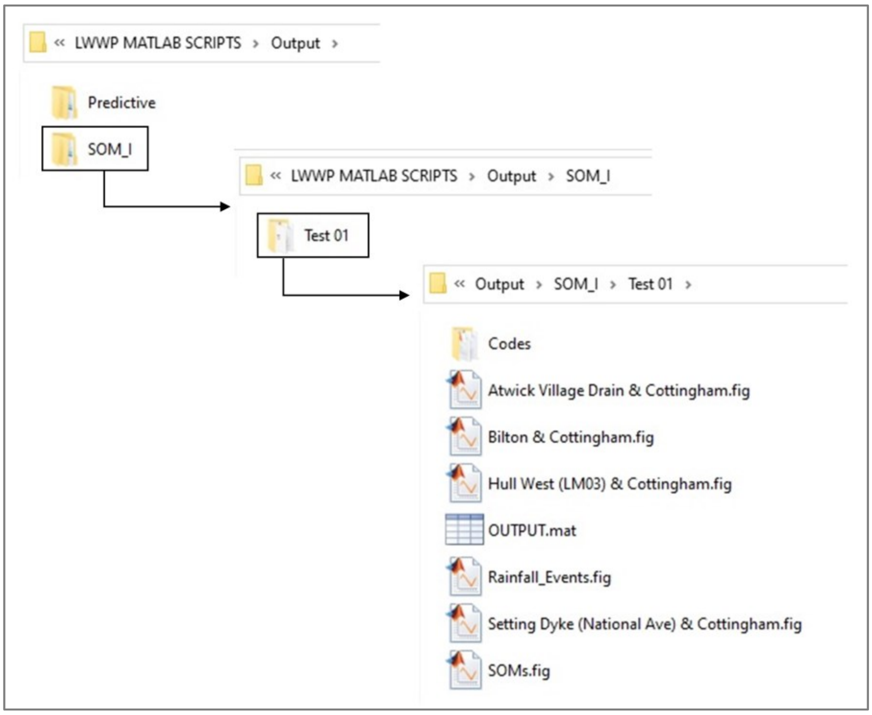

« LWWP MATLAB SCRIPTS › Output ›
Predictive
SOM_I
« LWWP MATLAB SCRIPTS › Output › SOM_I
Test 01
« Output › SOM_I › Test 01 ›
Codes
Atwick Village Drain & Cottingham.fig
Bilton & Cottingham.fig
Hull West (LM03) & Cottingham.fig
OUTPUT.mat
Rainfall_Events.fig
Setting Dyke (National Ave) & Cottingham.fig
SOMs.fig

Figure 9

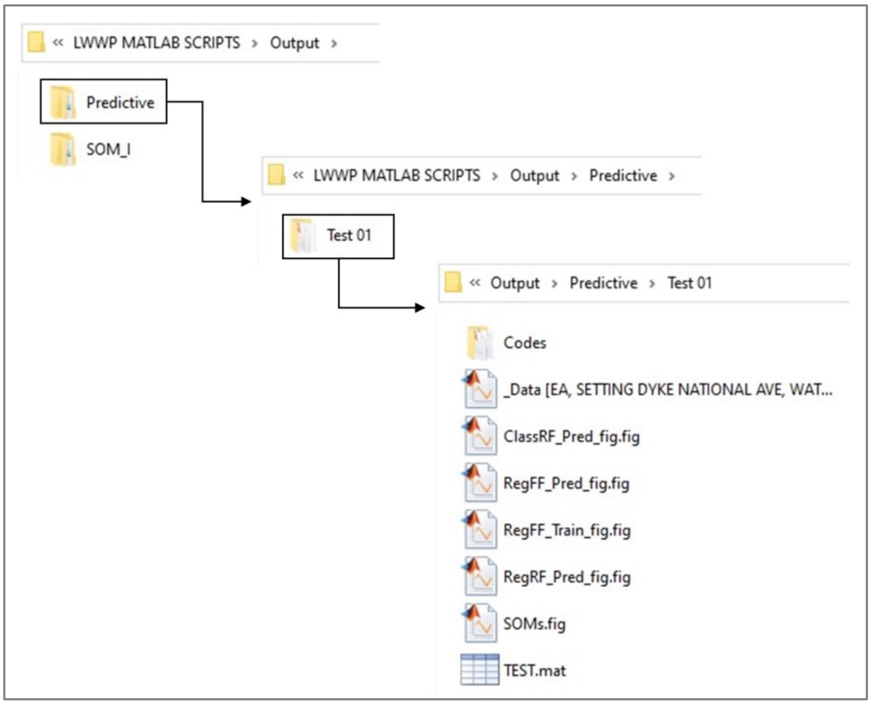

« LWWP MATLAB SCRIPTS › Output ›
Predictive
SOM_I
« LWWP MATLAB SCRIPTS › Output › Predictive ›
Test 01
« Output › Predictive › Test 01
Codes
_Data [EA, SETTING DYKE NATIONAL AVE, WAT...
ClassRF_Pred_fig.fig
RegFF_Pred_fig.fig
RegFF_Train_fig.fig
RegRF_Pred_fig.fig
SOMs.fig
TEST.mat

Figure 10

6. Data Cleansing/Filtering Model

The script for this model is located in folder *Codes for filtering time-series*. The part of script Filter_Data.m where the input parameters and specifications are set is presented below.

```
%% Data
DataID = 'TS7';       % Data ID in the DataList sheet
DataIn = 'RAW';       % Type of input data (end of input filename)
DataOut = 'FILTERED'; % Type of output data (end of output filename)

%% Remove part of time-series?
Tnan = {...
    '11-Jun-2015 10:00', '03-Jul-2015 13:15' ; ...
    ... '','' ; ...
    };

%% Level shift?
Shift = {...
    '12-Feb-2014 00:15', '05-Apr-2016 08:00',  '-6.63'; ...
    '17-Aug-2016 00:00', '04-May-2017 23:45',  '-6.63'; ...
    ... '','',  '' ; ...
    };

%% Remove spikes below Vmin and/or above Vmax?
Vmin = 1.2;
Vmax = [];

%% Filter time-series?
filt_coef = 3;             % no filter: '[]' (empty) or '1'
                           % recommended value: '3'

%% Smooth time-series?
smooth_func = '';          % no smoothing: '' (empty)
                           % 'moving'
                           % 'lowess'
                           % 'loess'
                           % 'sgolay'
                           % 'rlowess'
                           % 'rloess'
smooth_coef = 2e-4;

%% Plot and save as image?
plt = 'Yes';     % 'Yes' or 'No'
```

The data which is filtered is chosen by setting `DataID` to the ID of the time-series. This should match the ID in column ‘ID’ in Data List spreadsheet located in the data folder (see Figure 7).

`DataIn` defines the type of data which is filtered. Default is ‘RAW’; and the raw data should exist in the data folder (*Data Sample*). Its filename should match the one in column ‘FileName’ in the Data List spreadsheet and include ‘RAW’ in the end of file. Its extension should be ‘.csv’ as this code reads only CSV files. An example of input filename is “ERYC, BILTON, OPEN CHANNEL, WATER LEVEL, RAW.csv” (corresponding to time-series TS7 in Figure 7). Note that the raw data should also exist in mat-file format in the data folder because some other codes read data only in the mat format (with extension ‘.mat’).

`DataOut` defines the type of output file, which is ‘FILTERED’. The filtered data files will be saved in folder *Data Sample* with the filename obtained from ‘FileName’ column in Data List spreadsheet including ‘FILTERED’ in the end of file. This should not be changed because other codes will read filtered data with this name format. An example of output filename is "ERYC, BILTON, OPEN CHANNEL, WATER LEVEL, FILTERED.mat" (corresponding to time-series TS7 in Figure 7).

`Tnan` is used when a subset of data is required to be removed. It replaces the values in that part of time-series with NaN (Not a Number). First and second columns define beginning and end of time period which is required to be removed. If left empty (or commented by using three dots in the beginning of the line), this function will be disabled.

`Shift` is used to shift a part of time-series upward or downward if needed. First and second columns define beginning and end of time period which is required to be shifted, and third column define the shift value (if positive, the section will move upward, and if negative, it will move downward). If left empty (or commented by using three dots in the beginning of the line), this function will be disabled.

`Vmin` and `Vmax` are used when there are high frequency spikes in the time-series which are not easily removed by the using the filtering function in the next step. Data points with values below `Vmin` and above `Vmax` will be simply replaced by NaNs. If left empty, as [ ], this function will be disabled.

For data filtering, a 1-D median filtering model (*medfilt1* in MATLAB 2019b) is applied. `filt_coef` defines filter order. Recommended value is 3 for water level time-series. If it is set to 1, or left empty as [ ], no filter will be applied and the output time-series will be exactly the same as the input one.

An extra smoothing can be applied in this model. *smooth* function in MATLAB 2019b is used for this purpose. `smooth_func` defines the type of smoothing function and `smooth_coef` sets the span of the moving average. If `smooth_func` is left empty, smoothing will not be applied. Smoothing should be used only in special cases when data is too noisy, otherwise it is not recommended to bed used as it will cause losing a part of variations in time-series which might be important.

`plt` determines whether the output time-series will be plotted and saved, or not.

Figure 11 presents the result of filtering time-series TS7 (open channel water level at Bilton) using the model. For this test, the input parameters and specifications are set as shown in the above script. This figure is saved in folder *Data Sample* as a MATLAB figure (with extension '.fig'). In addition, the data of this analysis is saved in a mat-file (with extension '.mat') in the same folder.

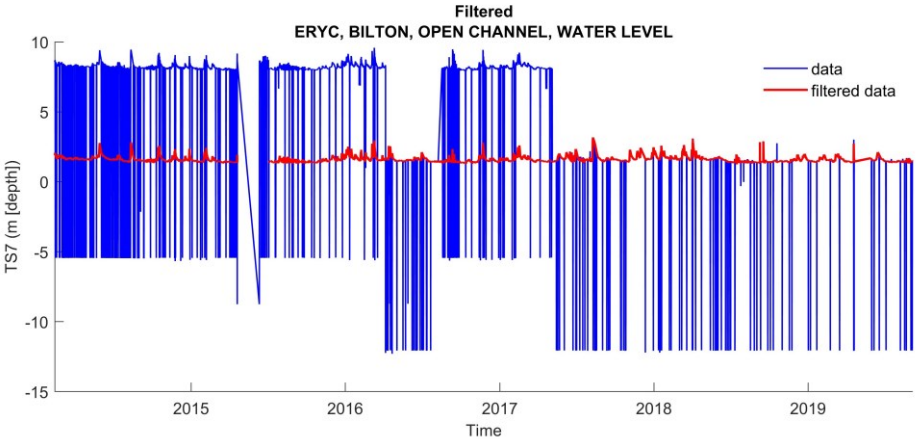


Figure 11

For other water level time-series, see their output mat-files saved in folder *Data Sample* (ending with 'FILTERED') to find out about parameters and specifications set to filter them. Note that for rainfall time-series (EA, COTTINGHAM, RAINFALL 15MIN), no filter is applied, and the 'raw data' is used in the SOM and Predictive Model analyses.

For further details of filtering time-series refer to LWWP Project Final Report.

7. Data Deseasonalising Model

Scripts for this model are located in folder *Codes for deseasonalising time-series*. To run the code, Deseasonalise_Data.m should be executed. The part of this script where the input parameters and specifications are defined is presented below.

```
%% Data
DataID = 'TS6';          % Data ID in the DataList sheet
DataIn = 'FILTERED';     % Type of input (end of input filename)
                         % 'RAW': deseasonilise raw data
                         % 'FILTERED': deseasonalise filtered data
DataOut = 'DESEASONALISED';  % Type of output (end of output file)

%% Deseasonalise time-series
Det_Method = 'poly';     % no deseasonalisation: '' (empty)
                         % 'poly' (recommended)
                         % 'smooth'
% -- 'poly'
degree = 0;              % use 0 for 'constant', or 1 for 'linear'
lower_percent = 0.20;    % lower part of data to determine baseline
seg_dur = 7;             % duration of each segment in days

% -- 'smooth'
smooth_func = 'rlowess';  % 'moving', 'lowess', 'loess',
                          % 'sgolay', 'rlowess', 'rloess'
nseg = 10;                % number of segments
smooth_coef = 0.3;        % span

%% Plot and save as image?
plt = 'Yes';     % 'Yes' or 'No'
```

The time-series which is deseasonalised is set by `DataID`. This should match the ID of the data in column ‘ID’ in Data List spreadsheet in the data folder (see Figure 7).

`DataIn` defines the type of data which is deseasonalised. There are two options: ‘RAW’ and ‘FILTERED’, if user would like to deseasonalise raw data or the data filtered by Filter_Data.m code, respectively. Input data file must exist in the data folder (*Data Sample*). In either case, input file name should match the one in column ‘FileName’ in the Data List spreadsheet (see Figure 7), while in the case of deseasonalising raw data, it should include ‘RAW’ in the end of file, and in the case of deseasonalising filtered data, it should include ‘FILTERED’ in the end of file (the latter is automatically done by Filter_Data.m code in the data filtering step). The input file format must be ‘.mat’ as this code reads only mat files. Examples of input filenames are “ERYC, BILTON, OPEN CHANNEL, WATER LEVEL, RAW.mat” or “ERYC, BILTON, OPEN CHANNEL, WATER LEVEL, FILTERED.mat” (corresponding to time-series TS7 in Figure 7).

`DataOut` defines the type of output file, which is set to ‘DESEASONALISED’. The output data files will be saved in folder *Data Sample* with the filename obtained from ‘FileName’ column in Data List spreadsheet including ‘DESEASONALISED’ in the end of file. This should not be changed because other codes will read deseasonalised data with this name format. An example of the output filename is “ERYC, BILTON, OPEN CHANNEL, WATER LEVEL, DESEASONALISED’.mat” (corresponding to time-series TS7 in Figure 7).

For removing seasonality from data, the unwanted seasonal components in the time-series are firstly modelled using curve-fitting or smoothing methods. The time-series is split into a number of segments and then at each segment polynomial curve fitting or smoothing functions are applied and then by combining the segments the seasonality is modelled and then removed. Therefore, there are two methods for deseasonalisation: using polynomial curve fitting functions, or smoothing functions, if `Det_Method` is set to ‘`poly`’ or ‘`smooth`’, respectively.

If ‘`poly`’ is chosen, ‘`degree`’ defines the order of curve fitting function. There are two options as shown in the script above. As mentioned above, the time-series is firstly split into a number of segments. ‘`seg_dur`’ determines the duration of a segment in days. In this method, a base line is defined at each segment, which represents the seasonal component of the time-series. ‘`lower_percent`’ defines the percentage of data points in the lower part of time-series to be used for determination of the base value. For more details about the process, see LWWP Project Final Report.

If ‘`smooth`’ is chosen, *smooth* function in MATLAB 2019b is used for removing seasonal components. In this case, `smooth_func` defines the type of smoothing function, `smooth_coef` sets the span of the moving average, and ‘`nseg`’ specifies the number of segments the time-series is divided into.

`plt` determines whether the output time-series will be plotted and saved, or not.

Figure 12 presents the result of deseasonalising time-series TS6 (open channel water level at Atwick Village Drain) using the model. For this test, the input parameters and specifications are set as shown in the above script. This figure is saved in folder *Data Sample* as a MATLAB figure (with extension '.fig'). In addition, the data of this analysis is saved in a mat-file (with extension '.mat') in the same folder.

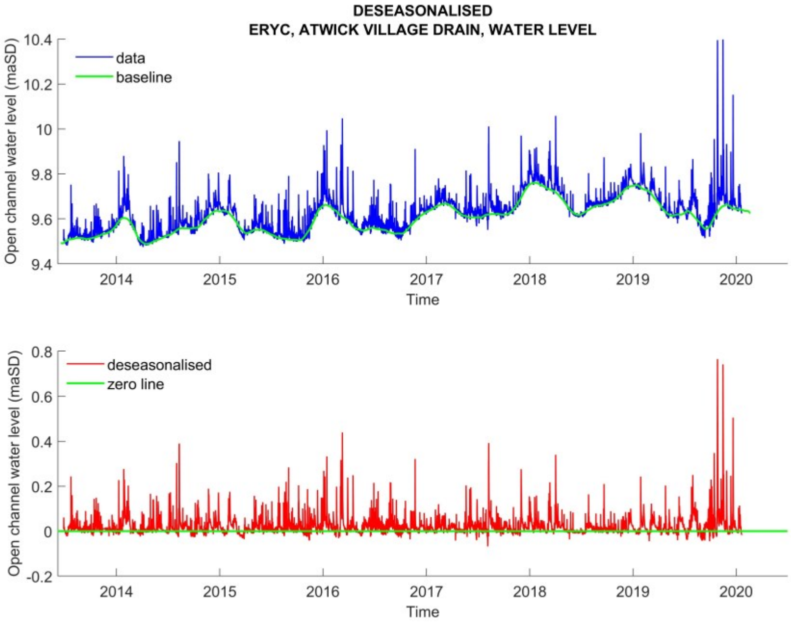


Figure 12

For other water level time-series, see their output mat-files saved in folder *Data Sample* (ending with 'DESEASONALISED') to find out about parameters and specifications set to create them.

For further details of removing seasonality from time-series refer to LWWP Project Final Report.

8. SOM Model

Relationships between rainfall and maximum water level in the open channel watercourses and the sewer network are explored by the SOM technique. This analysis is based on storm events. Firstly, rainfall events are detected, and two rainfall parameters are calculated for each event, i) total precipitation (mm) which is sum of rainfall values during the event, and ii) maximum rainfall intensity (mm/15mins) which is rainfall peak in the event. Then, maximum value of water level in the next `Nx` hours after rainfall event is calculated. These rainfall and water level parameters are fed into the SOM model.

For rainfall event detection, an automated system was developed which searches over the rainfall time-series, finds events with total precipitation or maximum intensity above a threshold, and then separates the events based on their distance in time, i.e. calculates the dry period between two successive events, and if it is below a threshold, they are combined into a single event, and if it is above the threshold, then they are kept as two separate events.

Scripts for this model are located in folder *Codes for Self-Organising Maps* (see Figure 4). To run the code, SOMS_I.m should be executed. The part of this code where the input parameters and specifications are defined is presented below.

```
%% Test Name/No
Test = 'Test 01';

%% Parameters (locations)
%  'RAW', 'FILTERED', or 'DESEASONALISED'
RainFile = 'Cottingham';      % rainfall tag in the Data List
LevelFiles = { ...            % water level tag in the Data List
    'Bilton'; ...
    'Atwick Village Drain'; ...
    'Setting Dyke (National Ave)'; ...
    'Hull West (LM03)'; ...
    };

LevelPlt = 'On';   % whether to plot water level time-series
                   % 'On' or 'Off'

%% Type of input data
RainSeriesType  = 'RAW';
LevelSeriesType = 'DESEASONALISED';

%% Start and end of time series used for analysis
tp_strt = '1-Jan-2010 00:00:00';
tp_end =  '1-Jan-2020 00:00:00';
timeStep = 0.25;  % hr

%% Observation window for water level
% water level Nx hours in the future is searched to find its maximum
Nx = 12; % hr

%% Parameters for Rainfall Event Detection Model
BasEv = 0;        % (mm) base value for rainfall event detection
                  % (default = 0)
tDry = 6;         % (hr) minimum dry period between two successive
                  % rainfall events
rain_ev_thresh = [0.5, 25];  % thresholds for rainfall intensity
                             % (mm/15min) and total precipitation
                             % (mm) to be used for event detection
                             % (events smaller than these will not
                             % be taken as events)
rainEvPlt = 'On';   % whether to plot events
                    % 'On' or 'Off'

%% SOM parameters (for options see SelfOrganisingMap function)
lqa = 0.02;
uqa = 0.98;
init = 'lininit';
algo = 'imp';
```

```
mpsiz = 'normal';
```

`Test` is used to give a name to the test, and this will be the folder name where the output files are saved (see Section 5 and Figure 9).

`RainFile` Is the name of rainfall data. It should match the name in column Tag in Data List spreadsheet (see Figure 7).

`LevelFiles` contains names of water level data used in the analysis. They should match the names in column Tag in Data List spreadsheet (see Figure 7).

`LevelPlt` specifies whether to plot water level time-series and save them as images in the output folder (‘On’ or ‘Off’).

`RainSeriesType` and `LevelSeriesType` determine the type of input data for rainfall and water levels. For example, if `'DESEASONALISED'` is chosen, the deseaosanalised time-series will be used in the SOM analysis. Note that the data files ending with the specified type, matching with column FileName in the Data List spreadsheet, and in the mat format (e.g. ‘YW, HULL WEST LM03, WATER LEVEL, DESEASONALISED.mat’) must exist in the data folder (*Data Sample*) before executing the code.

`tp_strt`, `tp_strt` and `timeStep` specify the start, end and time interval of the section/subset of time-series that user would like to use in the analysis. For instance, in the above code, a period of 10 years of data, from 2010 to 2020, with a time step of 15 minutes, is used.

`Nx` specifies the observation window for water level. As discussed above, firstly, rainfall events are detected with two rainfall parameters; and then, maximum value of water level in the next `Nx` hours after rainfall event peak is determined.

`BasEv`, `tDry` and `rain_ev_thresh` are parameters of the Rainfall Event Detection Model. `BasEv` is the base of rainfall time-series, which is set to zero. `tDry` (hr) is the minimum dry period between two successive rainfall events, and `rain_ev_thresh` contains thresholds for rainfall intensity (mm/15min) and total precipitation (mm) to be used for event detection. Events with total precipitation smaller than the first element of the array, and with rainfall intensity smaller than the second element, will not be taken as events.

To avoid bad scaling of the colorbar, the SOM model uses the function *quantile* to keep the colorbar properly by ignoring outliers in the input data. `lqa`, `uqa` are used to rescale the colorbar. They are set to a number between 0 and 1. `lqa` specifies the lower boundary (and is usually set to be between 0 and 0.1) and `uqa` specifies the upper boundary (and is usually set to be between 0.9 and 1.0).

`init`, `algo`, and `mpsiz` are some of the hyperparameters of the SOM model which specify the initialization method, the training algorithm and the map grid size, respectively. For more details about the hyperparameters of the model (including other hyperparameters), and the options, see SelfOrganisingMap.m script. Note that to run the model, toolbox '_*SOM Toolbox v2.1*' is required to be added to the MATLAB path as explained in Section 3.

Figure 13 presents two examples of the events detected by the Rainfall Event Detection Model, and Figure 14 shows the result of SOM analysis using setup and specifications shown in the above script.

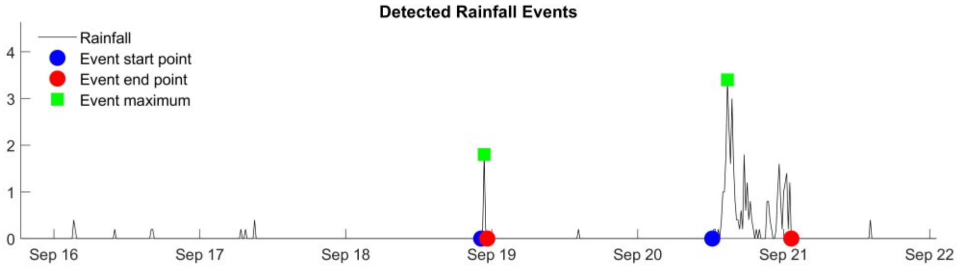


Figure 13

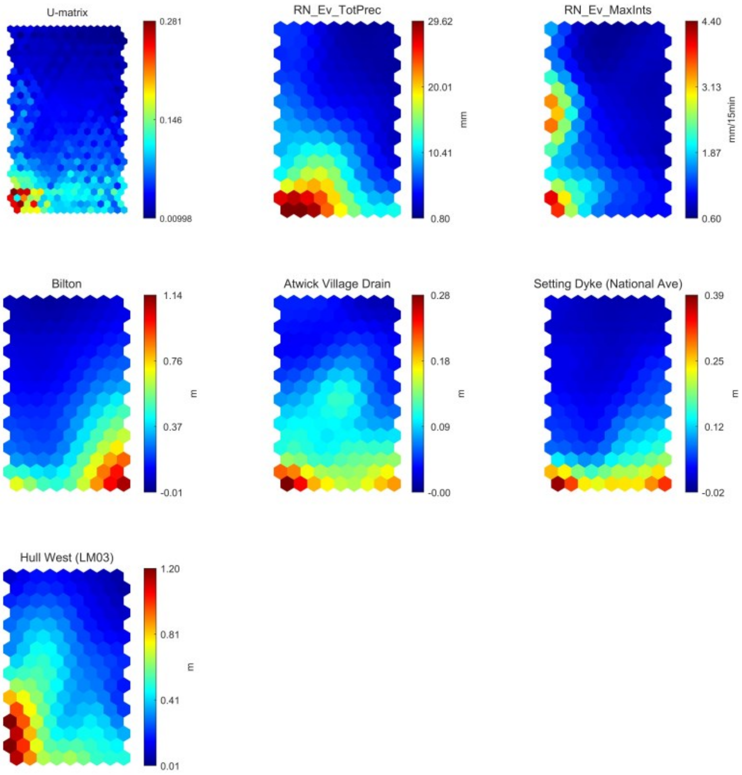


Figure 14

The SOM lattices in Figure 14 present two rainfall parameters ('TotPrec': total precipitation, and 'MaxInts': maximum intensity) at `Cottingham` and water levels at `Bilton`, `Atwick Village Drain`, `Setting Dyke (National Ave)` and `Hull West (LM03)`, as specified in RainFile and LevelFiles arrays. These two figure are saved in 'LWWP MATLAB SCRIPTS\Output\SOM_I\Test 01' as MATLAB figures (with extension '.fig'). In addition, other figures, output data, and executed code are all saved in the same folder.

For details of the analysis, refer to the output file 'OUTPUT.mat' saved in the output folder. For further details of SOM analysis of the LWWP data, refer to LWWP Project Final Report.

9. Predictive Model

In this analysis, output parameter, to be forecasted, is water level in a location specified by user, and input parameters could be a range of different parameters such as rainfall, water level in the same or other locations, groundwater level, etc, in the past, which are all specified by user. For details of the modelling process refer to LWWP Project Final Report.

Four machine learning techniques are available to be used in this model: SOM, RF_class, RF_reg and FF_reg. The last three ones, which are based on classification based Random Forests, regression based Random Forests and regression-based Feed-Forward Artificial Neural Networks, are the models used for water level predictions. SOM is also included as an extra tool to further investigate the relationships.

Scripts for this model are located in folder *Codes for predictive model* (see Figure 5). To run the code, PredictiveModel.m should be executed. The part of this code where the parameters and specifications are defined is presented below.

```
%% Test Name
Test = 'Test 01';

%% Models ('On' or 'Off')
Model.SOM       = 'On';    % use SOM?
Model.RF_class  = 'On';    % use Classification Random Forests?
Model.RF_reg    = 'On';    % use Regression Random Forests?
Model.FF_reg    = 'On';    % use Regression Feed-Forward ANN?

plt = 'On';    % Plot input/output time-series? 'On' or 'Off'

%% Define input and output parameters

% -- Input parameters (variable 'Inparam') -------------------------
%
% Definition: Observation window is the period over which an input
%             parameter is determined and extends from current time
%             to several hours in the past.
% Rows:
%             Each row in 'Inparam' is an input parameter.
% Columns:
%           - First column is name of data/location (should match
%             'Tag' in the Data List spreadsheet).
%           - Second column is type of data ('RAW', 'FILTERED',
%             or 'DESEASONALISED'). It should be capitalised and
%             corresponding data file should exist in the data
%             folder.
%           - Third column is type of parameter ('sum', 'max',
%             'mean', 'mag', 'slp').
%             mag  = magnitude of input parameter at X hours in
%                    the past (i.e. at the beginning of its
%                    observation window).
%             mean = average of input parameter over its observation
%                    window (past).
%             max  = maximum value of input parameter over its
%                    observation window (past)
%             sum  = sum of values of input parameter over its
%                    observation window (past)
%             slp  = gradient of profile of input parameter over
%                    its observation window (past)
%           - Fourth column is the size of observation window (i.e.
%             over how many hours in the past the value of input
%             parameter should be determined).
%           - Fifth column is cut-point for categorical parameters.
%             It defines the value at which classes of an input
%             parameter are defined, e.g. classes of water level
%             below and above a threshold (if it is left empty,
%             parameter is not set as a categorical parameter, but
%             as a numerical one).
% ------------------------------------------------------------------

Inparam = { ...
    'Cottingham'                  'RAW'      'sum'   '18'    ''; ...
    'Setting Dyke (National Ave)' 'FILTERED' 'slp'   '2'     ''; ...
    'Setting Dyke (National Ave)' 'FILTERED' 'mean'  '9'     ''; ...
 ...'Cottingham Willerby Hill'    'FILTERED' 'mean'  '3'     ''; ...
    };

% -- Output parameter (variable 'Outparam') ------------------------
%
% Definition: Prediction window is the period in the future over
%             which the output parameter is determined. It extends
%             from a time in the future to another time in the
%             future. For example, between 3 and 6 hours in the
%             future.
% Rows:
%             There is only one row, i.e. one output parameter.
% Columns:
%           - First column is name of data/location (should match
%             'Tag' in Data List spreadsheet).
%           - Second column is type of data ('RAW', 'FILTERED',
%             or 'DESEASONALISED').
%           - Third column is type of parameter ('mag', 'mean' or
%             'max').
%             mag  = magnitude of output parameter at X hours in
%                    the future. In the case of 'mag', only starting
%                    point of the prediction window (element in the
%                    fourth, see below) is taken into account.
%             mean = average of output parameter in the prediction
%                    window (future).
%             max  = maximum value of input parameter in the
%                    prediction window (future).
%           - Fourth and fifth columns indicate start and end point
%             of the prediction window from current time, i.e. time
%             distance of start and end of prediction window to
%             current time, for example 2 and 4 hours in the future.
%             In the case of prediction of magnitude of output
%             parameter (third column set to 'mag'), only first
%             point (fourth column) is taken into account).
%           - Sixth column is cut-point for the classification RF
%             model (RF_class), i.e. output parameter is defined as
%             a categorical parameter with classes of Positive and
%             Negative. Positive is when value of output parameter
%             is above this value and Negative is when it is below
%             this value. This value is used only for the
%             Classification RF model. For regression-based models
%             (Regression RF and Regression FF) it is ignored.
%           - Seventh column: for the Regression RF and FF models,
%             performance metrics (1-RMSE and NSE) are calculated
%             not only for the entire water level profile in the
%             predicted section, but also for a subset of data above
%             a threshold (e.g., -0.25 m), because water level data
%             is skewed due to the fact that most part of the
%             profile is at a constant water level which is around
%             the normal level of the channel. The value in the
%             seventh column defines this threshold.
% ------------------------------------------------------------------

Outparam = { ...
    'Setting Dyke (National Ave)'  'FILTERED' 'mag'  '3'  '3' ...
    '-0.1' '-0.25'};

%% Start and end of time series used for analysis
% -- entire period
tp_strt = '01-Jan-2012 00:00:00';
tp_end =  '30-Jun-2020 00:00:00';
timeStep = 0.25;  % hr
% -- period used as unseen data to be predicted after model training
t1_predict = '01-Mar-2018';
t2_predict = '20-Apr-2018';
```

`Test` is used to give a name to the test, and this will be the folder name where the output files are saved (see Section 5 and Figure 10).

`Model` determines which models to be used in the analysis. They can be activated or deactivated by setting this parameter to ‘On’ or ‘Off’.

`Plt` specifies whether to plot input/output time-series and save them as images in the output folder (‘On’ or ‘Off’).

`Inparam` and `Outparam` are used to set input and output parameters, respectively, and to specifiy the size of observation window, prediction window, etc. The details of how these two parameters are defined and specified are presented in the script below (as comments).

`tp_strt`, `tp_end` and `timeStep` specify the start, end and time interval of the section/subset of time-series that user would like to use in the analysis.

A large portion of the data (usually 90 to 95 % of the time-series) is used to train the models, and a smaller part of it (usually 5 to 10 % of it) is used as unseen data to be predicted by the trained models. `t1_predict` and `t1_predict` specify the start and end of the subset used for predictions.

Using the setup and specifications shown in the above script, the code is executed and the results of prediction of water level in Setting Dyke (National Ave) at 3 hours in the future, as specified in `Outparam`, using FF_reg and RF_reg models are presented in Figure 15. As defined by `tp_strt`, `tp_end`, `timestep`, `t1_predict` and `t1_predict`, the model is trained by the data of a period of about 8 years from 2012 to 2020, and then used to predict a period of about 50 days in 2018.

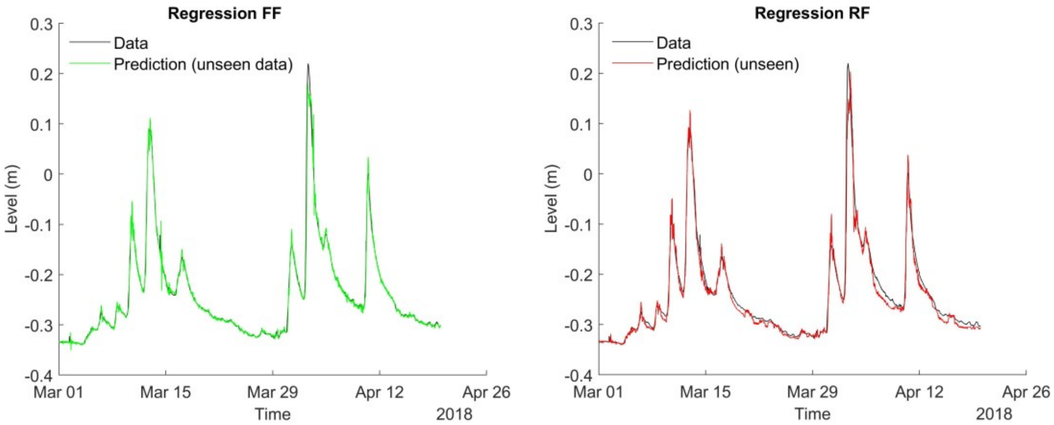


Figure 15

For details of this analysis, hyperparameters used for the models, and other information, refer to the output file ‘TEST.mat’ as well as the scripts saved in the output folder. For further details of the predictive modelling analyses of the LWWP data, refer to LWWP Project Final Report.

References

Reference LWWP Project Final Report