Evaluation of the MCLA in assessing the security of contingencies for the overload problem

October/2018

This document explains how to run the tool developed to assess the benefits from exploiting uncertainty forecasts over the TSO current practices of using deterministic forecasts of the system operating conditions, when applied to perform branch flow security assessment.

The following metrics are computed:

- Univariate Analysis
    - Location and dispersion metrics
    - Global and weekly security classification
    - Trade-off analysis to evaluate the impact of the used parametrization for security classification of a forecast
    - Metrics time evolution
    - Boxplot visualization for the most loaded situations
- Multivariate Analysis
    - Global and weekly security classification
    - Trade-off analysis to evaluate the impact of the used parametrization for security classification of a forecast
    - Metrics time evolution

1 Script Code

The tool hereby detailed is a script written in the R programming language [1]. The R version used in the development was 3.4 and the following R packages are required:

- **`xtable`** (version 1.7-4) and **`latex2exp`** (version 0.4.0) to export results into the latex document format;
- **`reshape2`** (version 1.4) for conversion between long and wide data format;
- ggplot2 (version 2.2.1) and **`gridExtra`** (version 2.3) for graphics rendering;
- **`latex2exp`** (version 0.4.0) to convert LaTeX math formulas to R's plot math expressions;
- **`lubridate`** (version 1.7.1) to deal with dates;
- **`doParallel`** (version 1.0.11) to perform parallel computation;
- **`DescTools`** (version 0.99.23);
- **`data.table`** (version 1.10.4-3) to deal with large .csv files.

This tool consists of a main folder, called "**`main_code`**" in Figure 1, which contains the relevant R scripts and a configuration file:

- **`main_functions.R`** is a script which contains all the functions to compute the proposed metrics.
- **`run_code.R`** is an auxiliar R script which runs **`main_functions.R`** and saves the results according to the definitions in **`config_file.txt`**.
- **`config_file.txt`** is a configuration file which may be edited to change such definitions as the name of the output folder and metrics to be computed.

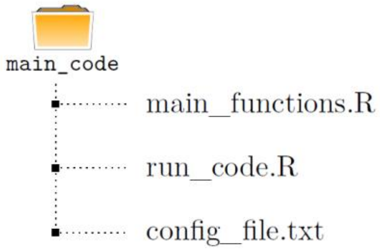
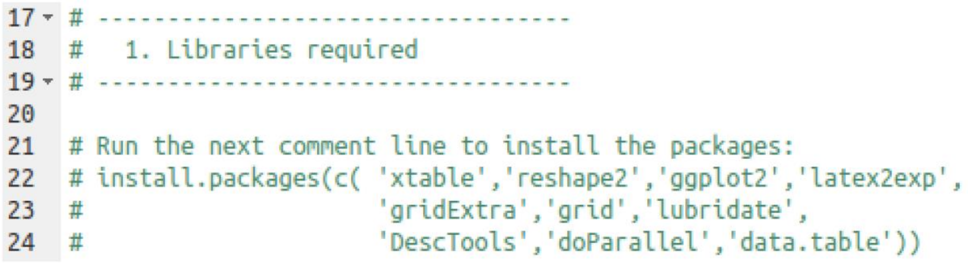


Figure 1 - Schematic representation of the tool

Remarks:

- The required packages mentioned above are automatically installed on the first run, if not installed already. In any case, you might want to run the following described command to ensure the latest versions of the packages are installed:
  - **`main_functions.R`** contains three commented lines (see Figure 2) which should be uncommented in the first run, in order to install the necessary R packages. Alternatively, the user can copy and run the following command:

    ```
    install.packages(c(’xtable’,’reshape2’,’ggplot2’,’latex2exp’,
    ’gridExtra’,’grid’,’lubridate’, ’DescTools’,’doParallel’,
    ’data.table’))
    ```
- In order to avoid format errors, the **`config_file.txt`** should be edited and saved using a text editor.



```
# ---------------------------------
#   1. Libraries required
# ---------------------------------

# Run the next comment line to install the packages:
# install.packages(c( 'xtable','reshape2','ggplot2','latex2exp',
#                     'gridExtra','grid','lubridate',
#                     'DescTools','doParallel','data.table'))
```

Figure 2 – Commands to uncomment in the first run

2 Input data

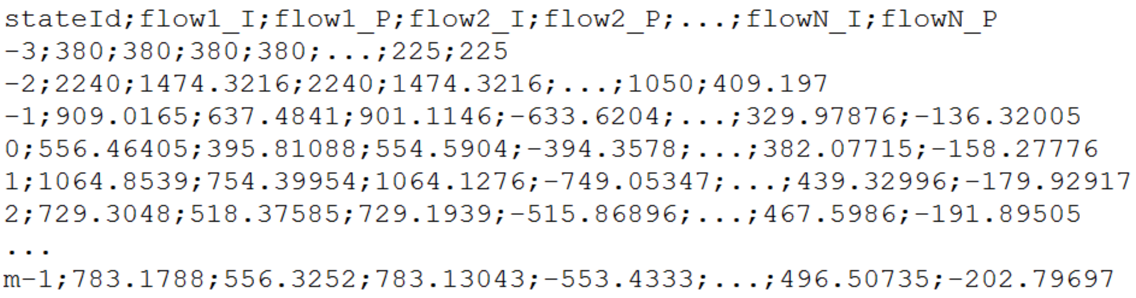
2.1 Analyzed data

For each specific timestamp YYYY-mm-dd HH:MM, a ".csv" file named "**`WF_YYYYmmdd_HHMM.csv`**" is used (example: **`WF_20170201_0030.csv`**). This CSV file is obtained using a MATLAB script developed by INESC TEC and is organized as follows (see illustration in Figure 3):

1. The first column identifies the state ID where:
   - State -3 identifies the line in the data with the nominal voltage of the transmission line (Un in kV).
   - State -2 identifies the line in the data with the permanent maximum limit values of each recorded operating condition ("Imax", in A, for electric currents and "Smax", in MVA, for active power flows).
   - State -1 identifies the line in the data with the SN load-flow results.
   - State 0 identifies the line in the data with the DACF load-flow results.
   - The remaining states (state 1, state 2, ...) identify the remaining ensemble members load-flow results. These ensemble members are the MCLA states created for the DACF base

case. Note that this states goes from State 1 to State *m*-1, with *m* being the number of ensemble members provided from the MCLA, including the DACF base case.

2. The remaining columns identify the recorded steady-state operating conditions, which can be the active power flow (P in MW) or the electric current (I in A), for each pre-selected transmission line flow (usually for post-contingency results but pre-contingency results can also be used).



```
stateId;flow1_I;flow1_P;flow2_I;flow2_P;...;flowN_I;flowN_P
-3;380;380;380;380;...;225;225
-2;2240;1474.3216;2240;1474.3216;...;1050;409.197
-1;909.0165;637.4841;901.1146;-633.6204;...;329.97876;-136.32005
0;556.46405;395.81088;554.5904;-394.3578;...;382.07715;-158.27776
1;1064.8539;754.39954;1064.1276;-749.05347;...;439.32996;-179.92917
2;729.3048;518.37585;729.1939;-515.86896;...;467.5986;-191.89505
...
m-1;783.1788;556.3252;783.13043;-553.4333;...;496.50735;-202.79697
```

Figure 3 – Contents illustration of each "WF_YYYYmmdd_HHMM.csv" file

Note 1: The code name used for these operating conditions is the one obtained from the files generated by the iPST online platform.

Note 2: The maximum permanent limit for active power flows ("Smax", in MVA) is assumed to be calculated from the maximum permanent limit for the electric current ("Imax", in A) and by using the line nominal voltage ("Un", in kV). Namely, by assuming that Smax = sqrt(3)×Un×Imax/1000.

2.2 Configuration data

The relevant configurations are defined in the **config_file.txt** file. In this file, the following parameters may be changed:

- **results_path**: path to save all the results.
- **data_path**: path with the input data to be analyzed.
- **do_parallel**: boolean value (TRUE or FALSE) which indicates if parallel computation is performed (use more than one CPU), typical value=TRUE.
- **Cores**: defines the number of CPU cores to use in parallel computation (ignored if do_parallel=FALSE), typical value=8. If Cores is greater than the available CPU cores then the maximum number of available cores is selected.
- **p-value**: p-value for marginal quantiles definition, typical value=0.05.
- **SM-value**: Security margin used to perform security classification with deterministic forecasts, typical value=0.05.
- **Maximum nr of ensembles**: Maximum number of uncertainty forecasting members, typical value=50.
- **Time Evolution Plots**: boolean value (TRUE or FALSE) which indicates if the metrics time evolution plot analysis is to be performed and saved, typical value=TRUE.
- **Minimum timestamps**: minimum number of measures to include each variable in the analysis, typical value=100.
- **TradeOff Analysis**: boolean value (TRUE or FALSE) which indicates if the tradeoff analysis is to be computed and saved, typical value=TRUE.
- **Initial timestamp**: initial timestamp of the analysis (in format YYYY-mm-dd HH:MM:00) which can include or not the input data timestamps.

- **`Final timestamp`**: final timestamp of the analysis (in format YYYY-mm-dd HH:MM:00) which can include or not the input data timestamps.
- **`Fan.chart.p.valuei`**: defines the four values to use for the fan chart (illustrated in Figure 4), typical value ={0.025;0.05;0.1;0.15}.
- **`Bootstrap.random`**: boolean value (TRUE or FALSE) which indicates if the random resampling is to be performed for the bootstrap method, typical value=TRUE. Block resampling with daily moving block is assumed if **`Bootstrap.random=FALSE`**.
- **`re-dispatch.costs.options`**: option to define if the estimate of the required re-dispatch is to be computed. Possible values:
  - 1: the re-dispatch estimation is not performed;
  - 2: only the required re-dispatch estimation is performed and all the other analysis is ignored;
  - 3: all the analysis, including the required re-dispatch estimation, is performed.
- **`re-dispatch.costs.SMs`**: defines the system margins to use in re-dispatch estimation, typical value=0;0.01;0.02;0.03;0.04;0.05;0.06.
- **`re-dispatch.costs.Q(1-p)`**: defines the quantiles to use in re-dispatch estimation, typical value=0.5;0.6;0.7;0.75;0.8;0.85.

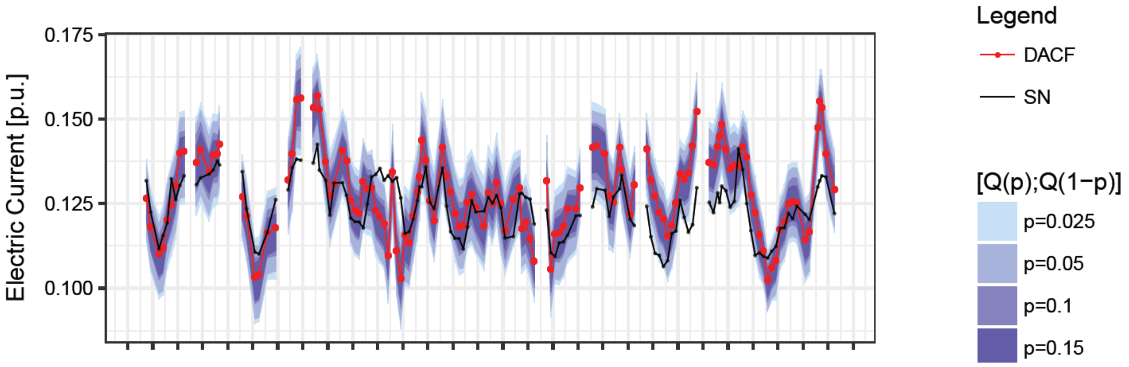
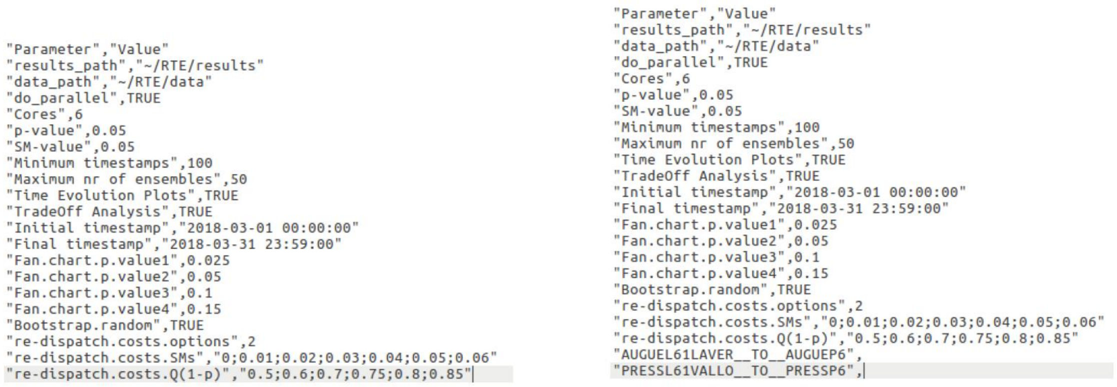


Figure 4 – Illustration of a computed fan chart (univariate analysis)

Finally, if the user wants to **remove from the analysis a specific set of line flows**, this can be performed by adding the code name of the line flow in the last lines of the **`config_file.txt`** (see Figure 5), **by default the analysis is performed by including all the inputted line flows**.

Example: If the user wants to remove the line flows whose code name in the input files is AUGUEL61LAVER__TO__AUGUEP6_I and PRESSL61VALLO__TO__PRESSP6_I then these code names must be included in the config_file.txt as illustrated in Figure 5.



```
"Parameter","Value"
"results_path","~/RTE/results"
"data_path","~/RTE/data"
"do_parallel",TRUE
"Cores",6
"p-value",0.05
"SM-value",0.05
"Minimum timestamps",100
"Maximum nr of ensembles",50
"Time Evolution Plots",TRUE
"TradeOff Analysis",TRUE
"Initial timestamp","2018-03-01 00:00:00"
"Final timestamp","2018-03-31 23:59:00"
"Fan.chart.p.value1",0.025
"Fan.chart.p.value2",0.05
"Fan.chart.p.value3",0.1
"Fan.chart.p.value4",0.15
"Bootstrap.random",TRUE
"re-dispatch.costs.options",2
"re-dispatch.costs.SMs","0;0.01;0.02;0.03;0.04;0.05;0.06"
"re-dispatch.costs.Q(1-p)","0.5;0.6;0.7;0.75;0.8;0.85"
```

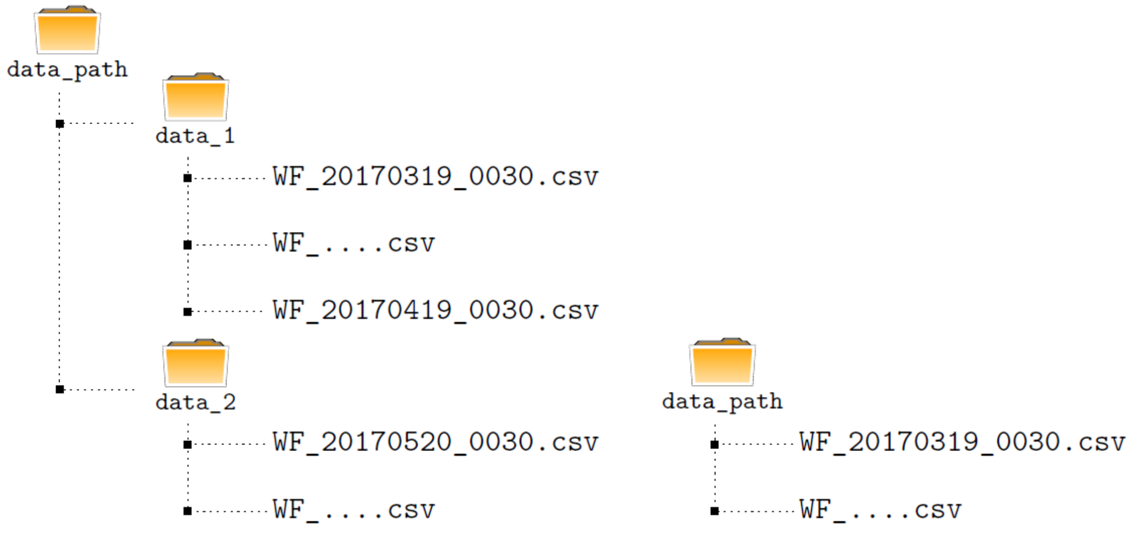
(a) The analysis is performed by including all the inputted line flows.

```
"Parameter","Value"
"results_path","~/RTE/results"
"data_path","~/RTE/data"
"do_parallel",TRUE
"Cores",6
"p-value",0.05
"SM-value",0.05
"Minimum timestamps",100
"Maximum nr of ensembles",50
"Time Evolution Plots",TRUE
"TradeOff Analysis",TRUE
"Initial timestamp","2018-03-01 00:00:00"
"Final timestamp","2018-03-31 23:59:00"
"Fan.chart.p.value1",0.025
"Fan.chart.p.value2",0.05
"Fan.chart.p.value3",0.1
"Fan.chart.p.value4",0.15
"Bootstrap.random",TRUE
"re-dispatch.costs.options",2
"re-dispatch.costs.SMs","0;0.01;0.02;0.03;0.04;0.05;0.06"
"re-dispatch.costs.Q(1-p)","0.5;0.6;0.7;0.75;0.8;0.85"
"AUGUEL61LAVER__TO__AUGUEP6",
"PRESSL61VALLO__TO__PRESSP6",
```

(b) A specific set of flows is removed from the analysis.

Figure 5 – config_file.txt illustration with and without variable selection

About **`data_path`:** The ".csv" files named "WF_YYYYmmdd_HHMM.csv" must be in a common folder and the code must receive the path for this common folder. The R script considers all the ".csv" files inside the **`data_path`** folder as input data. Figure 6 illustrates situations supported by the code.



(a) Data organized by sub-folders.

(b) Data without sub-folders.

Figure 6 – Input data folder illustration

Note 1: It is assumed that the last folder name of the **`data_path`** identifies the name of the analyzed contingency (because this analysis is usually performed for post-contingency results).

Note 2: When reading the data, the script will first search for a **`RData`** structure in the root of the **`data_path`** folder having the same specified Initial and Final timestamps (this last information is obtained from the name of the **`RData`** file). If this **`RData`** structure is not found, the script will go on with the task of reading data and will finish this task by saving a **`RData`** structure with the data (in the root of the **`data_path`** folder). On the contrary, if the **`RData`** structure is found, the script will read the data directly from the **`RData`** structure and, therefore, a huge time saving is obtained in the task of reading data.

3 Output data

As mentioned before, the output of the script is saved in the **results_path** defined by the user. Figure 7 illustrates the structure of this folder and the possible outputs, according to the configuration file.

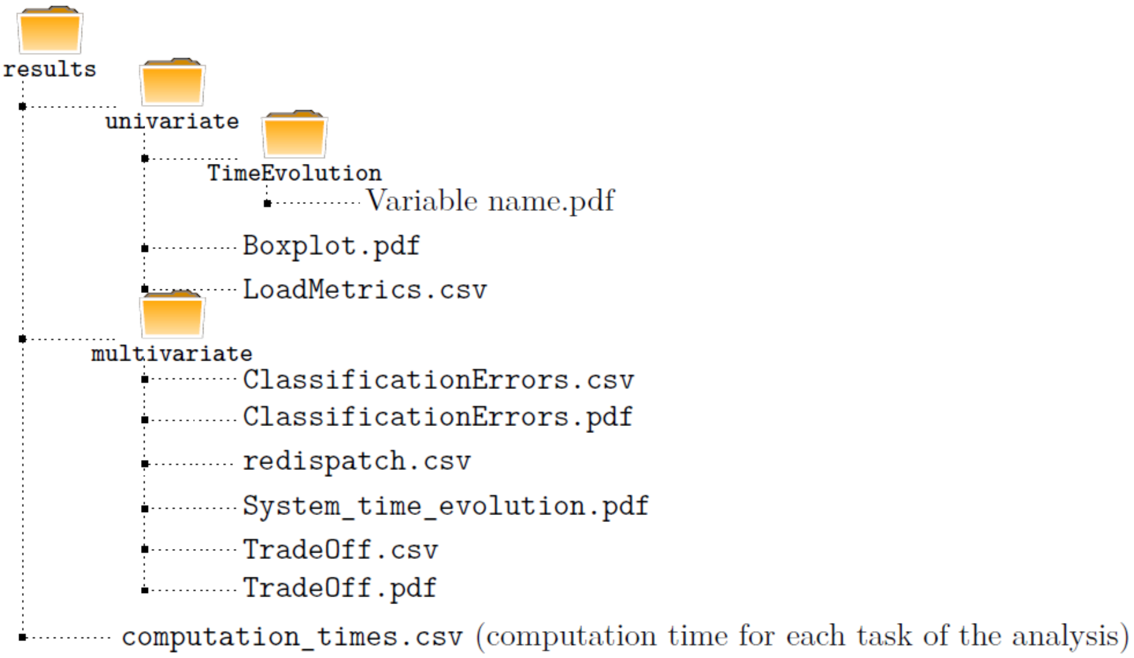


Figure 7 – Schematic representation of the results folder

4 First Use of the Code

In order to use this tool it is essential to have the **main_functions.R**, **run_code.R** and **config_file.txt** files in the same folder, named **path_tool**.

On the first use, the user needs to follow the following steps:

1. change the **path_tool** in **run_code.R**
2. change the **data_path** and **results_path** in **config_file.txt**
3. upgrade the required packages if there is an error related to the packages versions (see the above remarks in section 1).

After these procedures, the user just needs to execute the file **run_code.R** and all the results will be generated and saved, according to the specified parameters.

The tool runs in Linux or Windows operating systems. However, if Windows operating system is being used, in path definition it's mandatory to replace '\' by '/'.

5 References

[1] R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.