V0.43

# FDD Interface User Guide


This document is under periodical revisions, contents and screenshots may change.

## Table of Contents

**1 Introduction....3**
**2 Getting Started....3**
**3 H2G connections....4**
3.1 To get data....4
3.1.1 The “Nodes” tab....5
3.1.2 The “Files” tab....5
3.1.3 Operations....6
3.1.4 Making groups for ‘*sum*’ or ‘*swap*’ operations....7
3.1.5 Other settings....8
3.1.6 Controls at the bottom....9
3.1.7 Important notes....10
3.2 Creating models and events outputs....10
**4 Monitoring interface....11**
4.1 Control Area....11
4.2 Visualization Area....13
4.2.1 Previous results....16
4.2.2 Warnings....16
4.3 Scheduling of model executions.....18
**5 Modelling interface....20**
5.1 Modelling Area....20
5.2 Management Area....22
**Annex A: data input format....24**
**Annex B: Errors....25**

# 1 Introduction

The Fault Detection and Diagnosis (FDD) web application is a client application that uses some FDD algorithms available as web services. These web services are hosted in an external server, property of the UdG.

This web client allows the end-user to upload their own CSV files containing data, create a model, project data over the model, and show the results. Mainly, the web application is composed by 2 interfaces: modelling and monitoring. Basically, monitoring consists in continuous comparison of acquired variables against normal operating conditions of a system.
Next sections explore a brief guide to use these interfaces with a practical example.

**NOTE**: This application is currently under development. Disposition of controls and colours can vary during the development stage.

**IMPORTANT**: it is convenient to know and understand the input data format, since an error in the format may cause subsequent errors with the wrong conclusion "the application does not work". See Annex A.

# 2 Getting Started

Currently, to enter the application, the user must be identified by an ID and password. Thus, previously, the user must fill the registration form. When the information is sent, the administrator will be notified by email in order to enable the account before the user can use the application.

NOTE: Don't forget the password, since the password reset/reminder does not work yet.

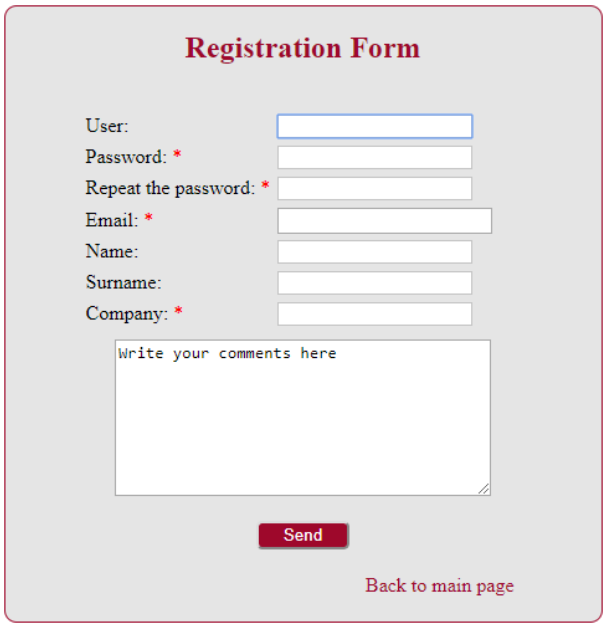



Figure 1 Registration form

After login, the user is addressed to the monitoring page (see Section 4). Also, depending of the user permissions, the user will be able to access to the modelling interface.

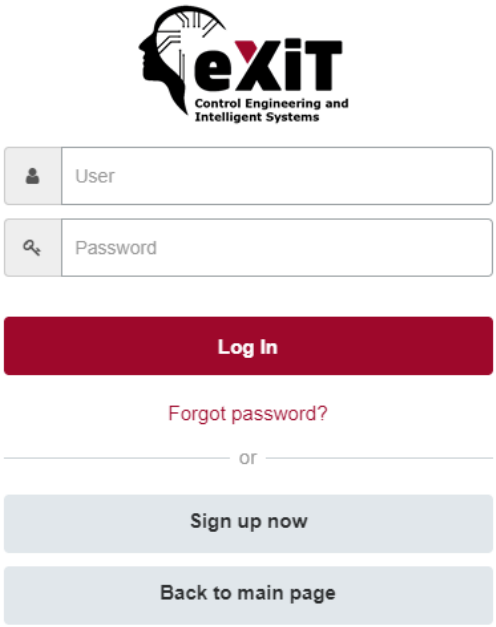



Figure 2 Login window

# 3 H2G connections

From the forecasting or modelling interface it is possible to visualize a modal interface that will allow us to do some basic operations with the H2G core. These are:

- To get data.
- To create models.
- To create timeseries outputs.

## 3.1 To get data.

It is possible to get data from the “Nodes” and “Files” tabs (Figure 3 and Figure 4). Next subsections describe the available operations.

### 3.1.1 The “Nodes” tab

Mainly, the “Nodes” tab (Figure 3) will be the one we must use for the first time. It allows selecting the variables and operations, the start and end time, the sampling time, and the name of the resulting CSV file. If the operation is done successfully, as a result it will be created two files, the data file and a data configuration file, containing the list of variables and settings. It is very important to get this file, since it will allow to configure the automatic execution of models.

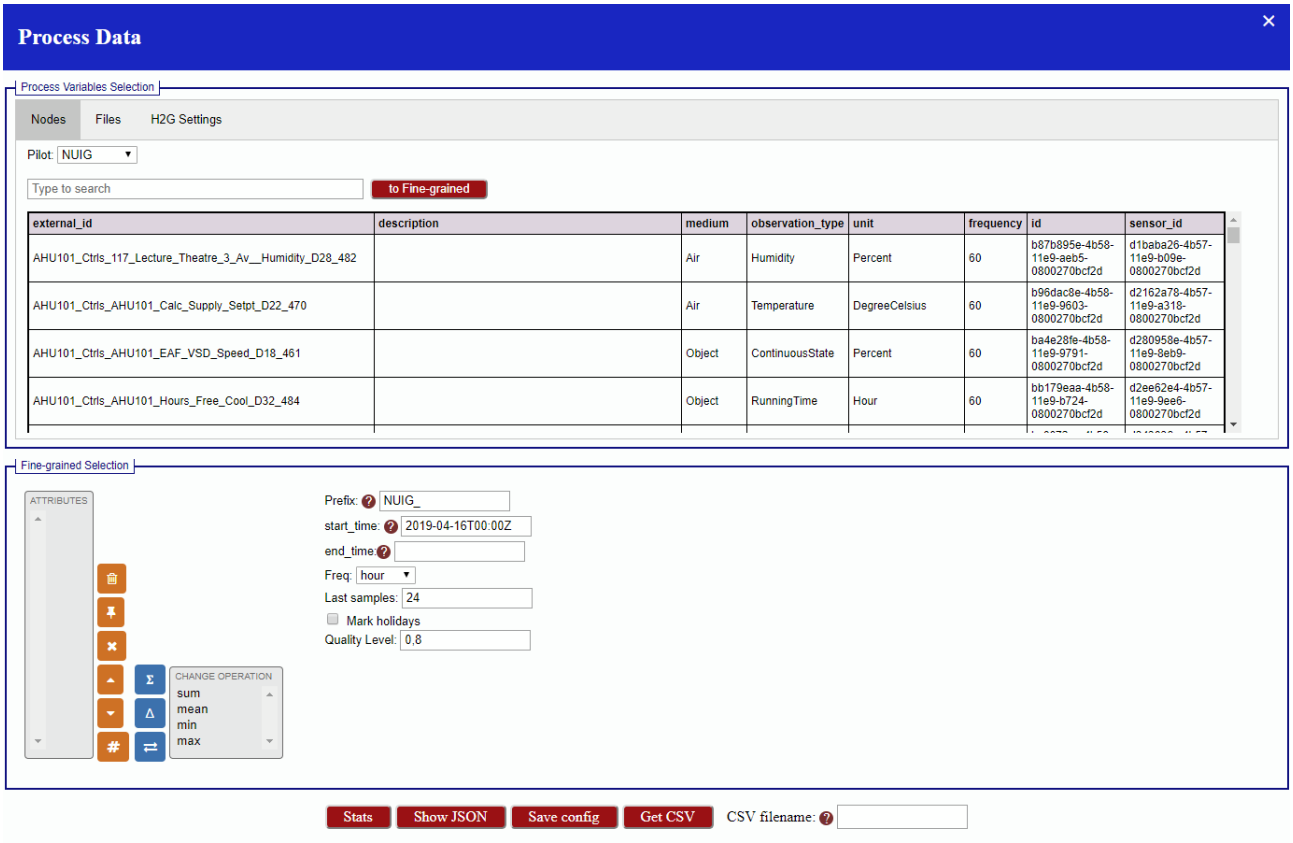



Figure 3 Getting data from the "Nodes" tab.

### 3.1.2 The “Files” tab

Because the settings of a list of variables could be a long and tedious task, it exist the “Files” tab. Through this tab, it is possible to get data selecting a model or a data configuration file previously created. Here the user can easily modify their configurations for the created models.

NOTE: if the user changes the data configuration file, it may be necessary to create a new model if the variables are different from those used by the model.

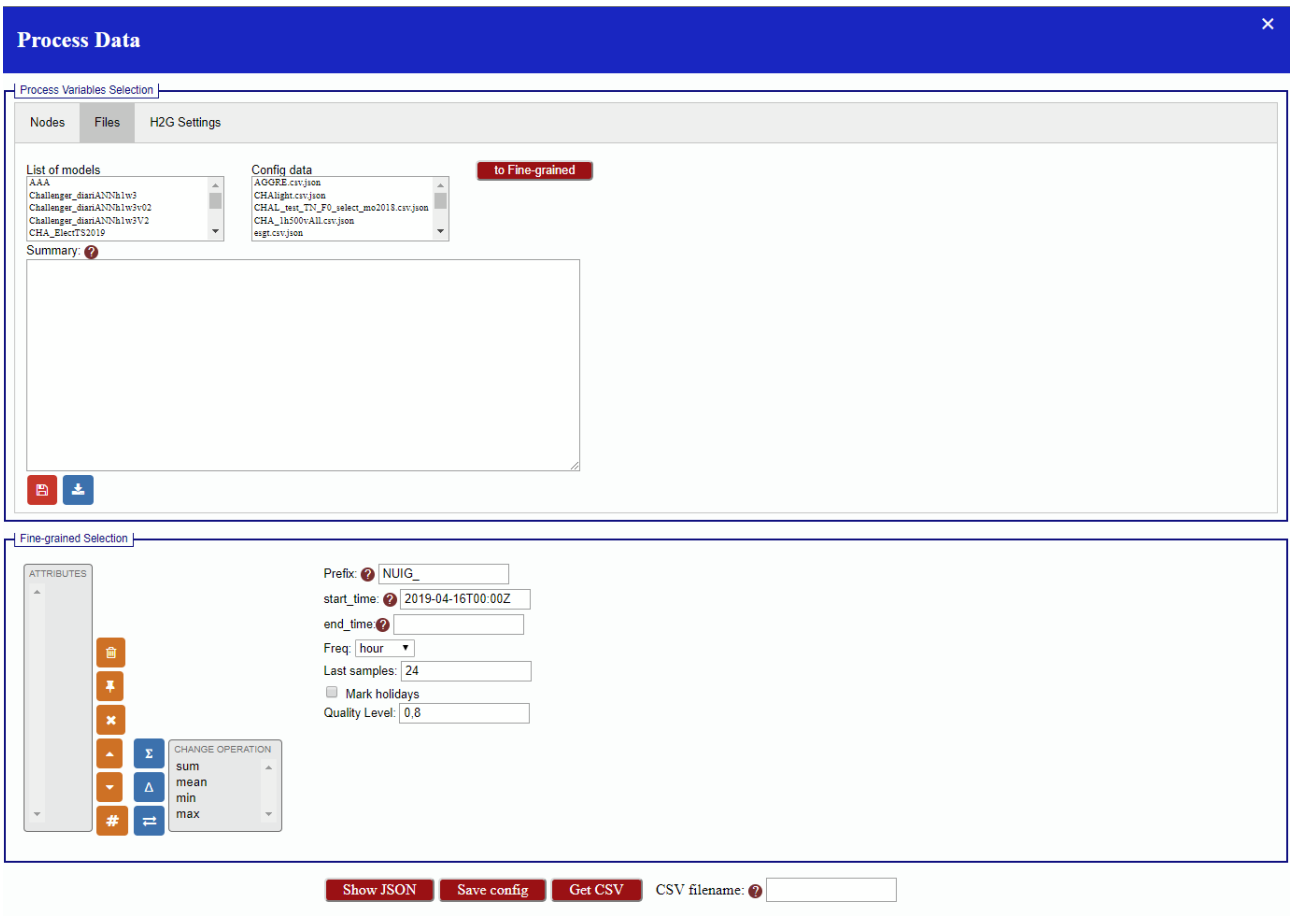



Figure 4 Getting data from the "Files" tab.

## 3.1.3 Operations

At the bottom of both tabs there is a list with the selected variables (Figure 5) and, on the right, there are some buttons to manage the list and to set each variable individually.

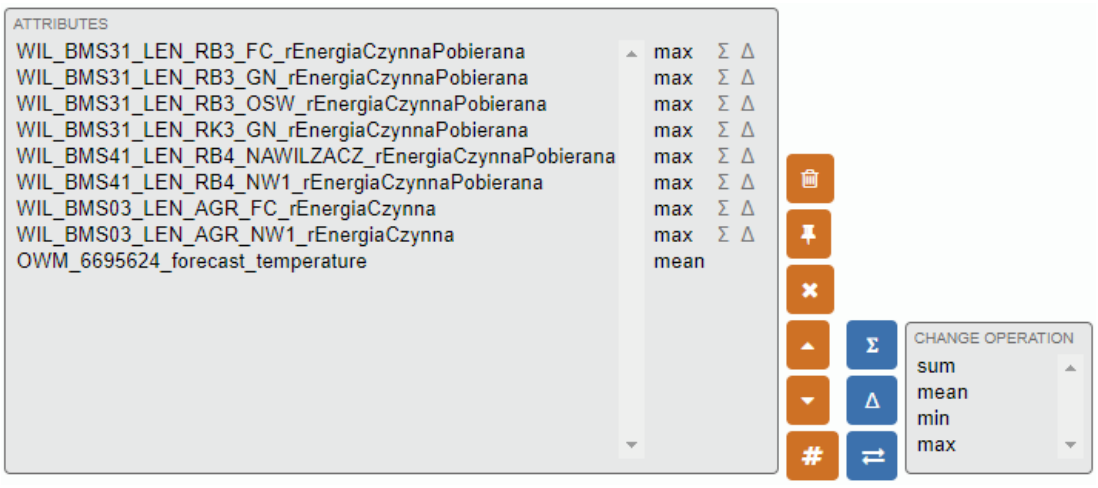
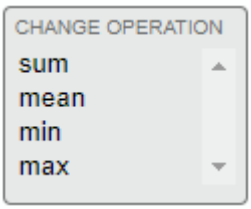



Figure 5 List of variables and available operations

To manage the list there exist some buttons in orange.

| | |
|---|---|
|  | Remove all the elements from the table |
|  | Select all the elements of the table |
|  | Remove the selected element from the table |
|  | Move up the selected element. Useful to make groups. |
|  | Move down the selected element. Useful to make groups. |
|  | Insert a separator. Useful to make groups. |

The operations performed by the H2G core are documented in the H2G API documentation. It is necessary to assign one operation to each variable, selecting the attribute and then one of the four operations. By default the operation is “max”.




In addition, there are available other useful operations performed by the UdG server. These are selectable with the blue buttons.

| | |
|---|---|
|  | Sum variables. The CSV file will contain an extra column named “S”. The user have to select individually all the attributes to sum, and then select this button. It will appear the Sigma symbol. |
|  | The resulting value will be the difference between samplings. Useful when the measures are meter readings. It is necessary an extra value at the beginning of the desired period. |
|  | Swap operation. It is necessary to have 2 attributes using this operation. The value will be the first attribute by default, but if this does not exist, the value will be replaced by the second attribute but using the name of the first one. |

### 3.1.4 Making groups for ‘*sum*’ or ‘*swap*’ operations

It is possible to configure different groups of variables to sum or swap (blue buttons). To do this it is essential to separate the groups in the list using the separator. For example, let’s consider the case of Figure 6. This configuration produces 4 attributes, although the resultant CSV will contain 5 variables. These are:

1. The variable 2.
2. The variable 3.
3. The variables 4, 5 and 6 use the ‘sum’ option, so this 3 variables generate a unique attribute named by default Agg1.
4. The variable 7 will be the fourth attribute. If some value it was no present in the BEMserver it will be filled by the value of the variable 8, if it exists, since they use the "swap" operator. The CSV also contain this last variable.

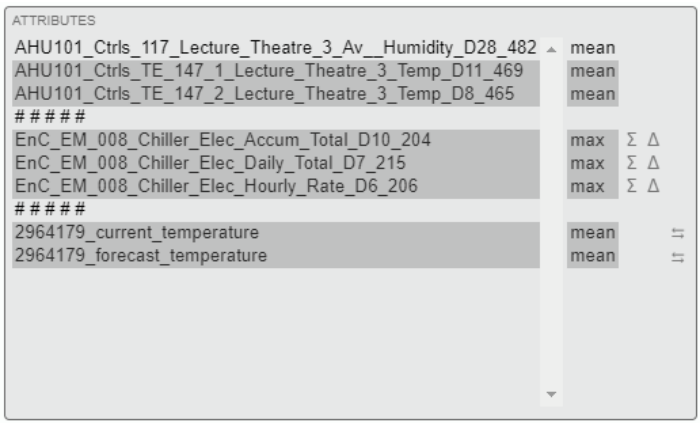



Figure 6 Example of configuration

## 3.1.5 Other settings

On the right side of the operations controls, there are some additional inputs to set up data collection (Figure 7).

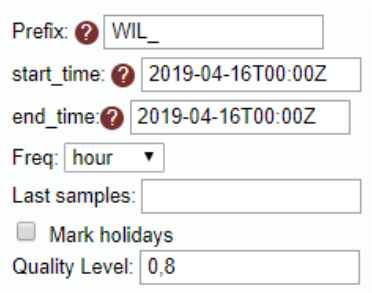



Figure 7 Aditional set up for data collection

**Prefix**: it could be necessary to add a prefix to the tag names. Check the names in the list.

**Start_time**: initial timestamp to download data. It must have the same format as indicated in the Figure 7.

**End_time**: final time to download data. This value is optional if a value is entered in the "Last samples" control. See final note.

**Freq**: sampling time.

**Last samples**: it indicates how many samples to download, from the beginning of the day (00:00Z).

**Mark holidays**: it will create an additional column (called '*Holidays*') indicating 0 (working days) or 1 (non-working days). National holidays are based on information from www.timeanddate.com.

**Quality Level**: it is the minimum quality level to accept the value of the attribute, otherwise it will be discarded.

### 3.1.6 Controls at the bottom

At the bottom of the tabs there are different controls listed below:

**Stats**: it calls the API *stats* for the selected variable. It is useful to explore the existence of this variable in the BEMserver.

**Show JSON**: it shows the resulting JSON according to the chosen configuration. Useful for developers.

**Save config**: it saves the configuration file (JSON) without downloading the data. It is useful when the user wants to change the name of the resulting aggregated variable, by default Agg*N* (see third attribute obtained in the example of Section 3.1.4). To do this, select the file in the "File" tab, and change the name in the "Summary" control (Figure 8). To be effective this choice the download must be done from this tab. Use the buttons under the "Summary" control to save the modified configuration file and/or download new data.

**Get CSV**: to download the data as a CSV file. A JSON configuration file will also be saved.

**CSV fieldname**: the name of the resultant CSV file, without the ".csv" extension.

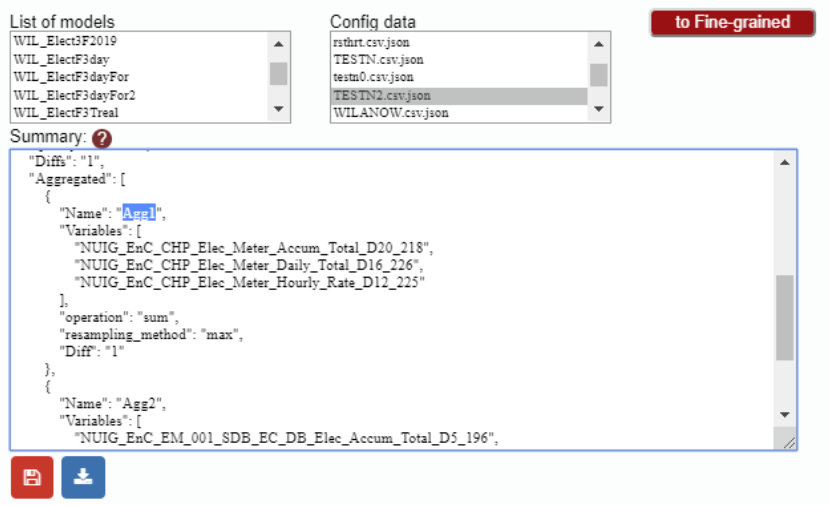



Figure 8 Changing names for aggregated variables.

### 3.1.7 Important notes

Please, consider:

- Only selected attributes will be downloaded.
- Separators must not be selected.
- If the “difference” operator is used within a group, all the variables must be marked with the symbol Δ (see example in Figure 6).
- If both timestamps are introduced (start and end) these values will be used to download data. If no value is introduced in *end_time*, the start and end time will be calculated according to the ‘*last samples*’ value.
- Finally, a JSON containing the settings will be created. This file will only contain the selected variables. This file will be attached to the model during its creation.

## 3.2 Creating models and events outputs.

To push results to the H2G core it is mandatory to create (register) first a model, and then create an event output for this model. The H2G provide the API to do it, but the tab “H2G Settings” (Figure 9) facilitates the user a way to do it. Here there are the following controls:

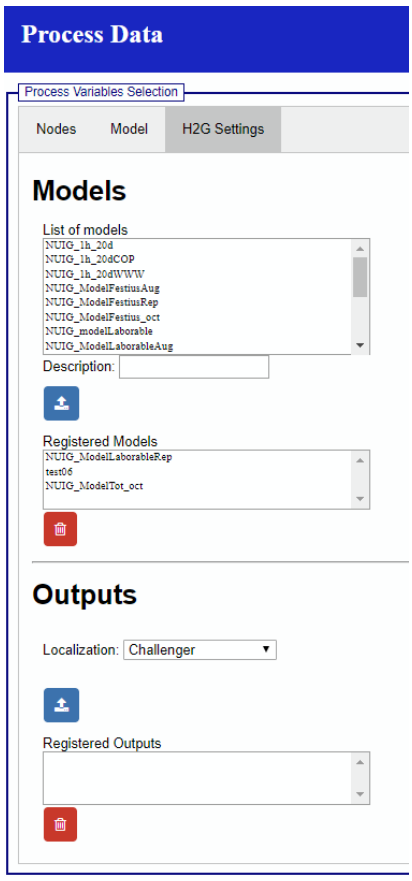


Figure 9 The H2G settings tab

**Models**

List of models: they are shown the built local models.

Description: this is a field to enter some description regarding the selected model.

Blue button (register model in the core):

Registered Models:

Red button (delete model from core):

**Outputs**

Localization: choose a localization.

Blue button (register output in the core):

Registered Outputs: shows the event outputs created in the H2G core for a selected model.

Red button (delete output from core): remove the selected output from the H2G core.

# 4 Monitoring interface

The monitoring page (Figure 10) is the main interface of the web application. It is accessible to all users and it allows to monitor new set of observations to detect abnormal operating conditions (faults/failures in sensors/equipment or user misbehaviours).

Visually, this interface is organized in two areas separated vertically. The left area correspond to the control of the interface, and the right one corresponds to the visualization of the data.

## 4.1 Control Area

The steps to start the monitoring (currently on manual request) are enumerated in this area. Next, these steps are reviewed in more detail.

**1. Upload a file**

This first step allows the user to upload a CSV file containing data for projection.

Another option is to get data online (see section 3.1)

These steps are optional, if the file has been previously uploaded.

**2. Select a file**

Here the user chooses one of the uploaded files for projection, between the available files shown in the '*Files uploaded*' control.

### 3. Select a model

Here the user chooses one of the created models for projection, between the available models shown in the list of models.

### 4. Press the button

Finally the user can initialize manually the projection of data. For this, the user have to select a file containing data and the corresponding model, and finally press the button “Calculate”. The server will return the results and they will displayed in the charts.

### Options

In addition, there are two options that can be selected before the execution of the previous actions. These are:

***Generate report***: it will open a report in the browser with the detected faults, if any.

***Send events to H2G***: if a fault was detected, the event will be sent to the H2G core, but only if the model and event were previously registered (see section 3.2).

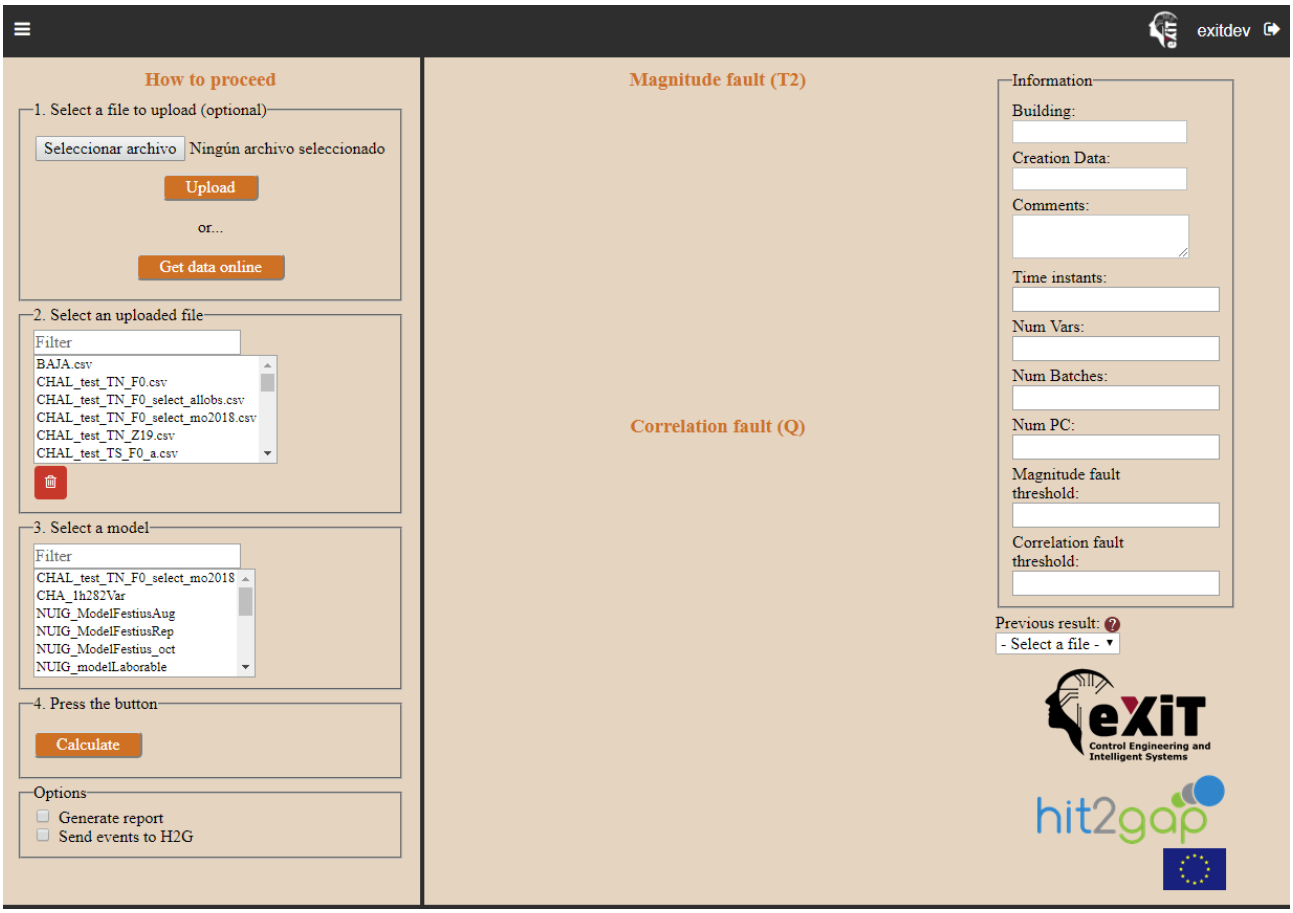



Figure 10 Monitoring interface

## 4.2 Visualization Area

The right side of the monitoring interface contains two monitoring charts to display the results, and some fields on the “Information” area show useful parameters about the selected model.

Regarding the charts, they represent simple statistic indexes $T^2$ (magnitude fault) and Q (correlation fault). These indexes are used to detect abnormal behaviours, represented as a dot over red areas (Figure 11). Thus, the red areas represent that one of the thresholds (or both) is overpassed, identifying a deviation of the new observation with respect to the model.

After detection, clicking on any of these dots on red area displays the contribution analysis (Figure 12). This new window allows to analyse the contributions of original variables to the statistics that detected the deviations. Also, it is possible to isolate variables responsible of such misbehaviour (Figure 13).

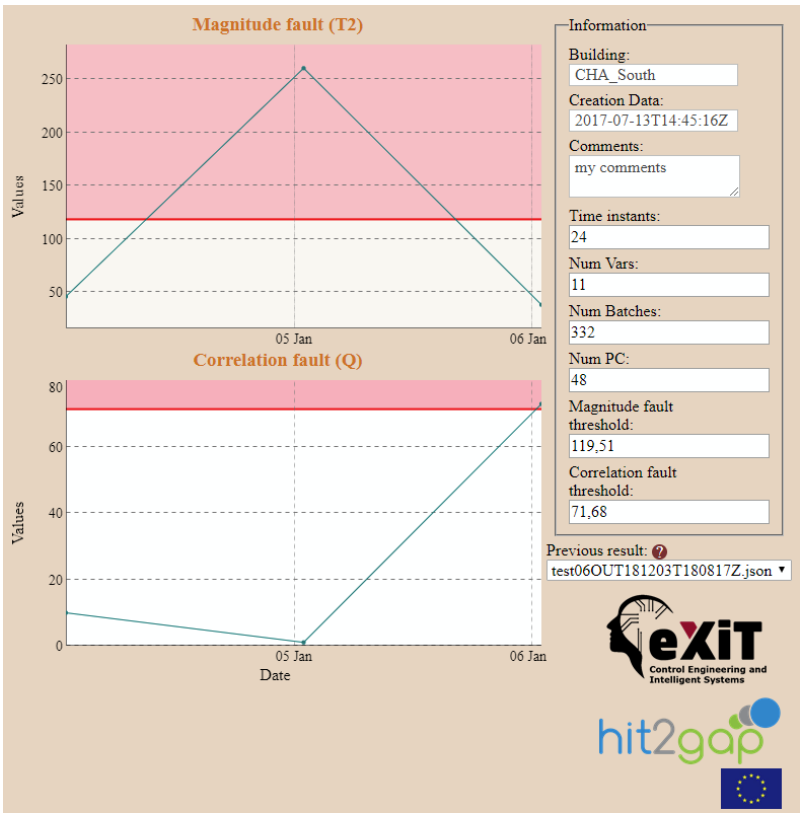



Figure 11 Visualization area detecting abnormal behaviour (red zones)

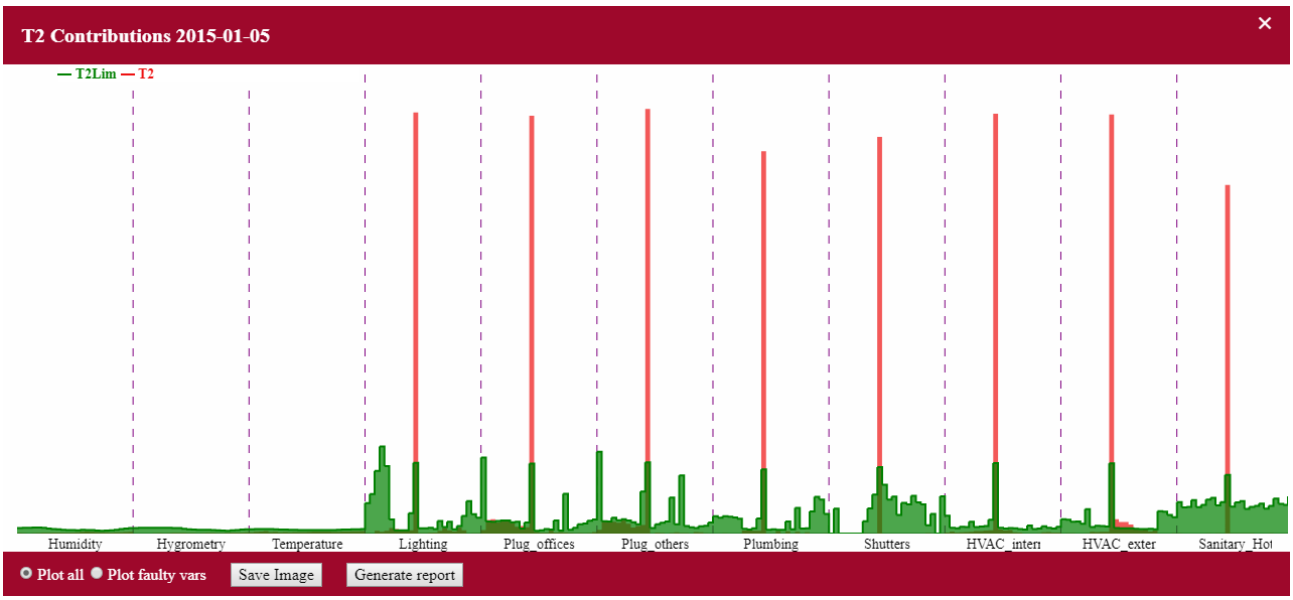



Figure 12 Contribution analysis of an observation with faulty variables

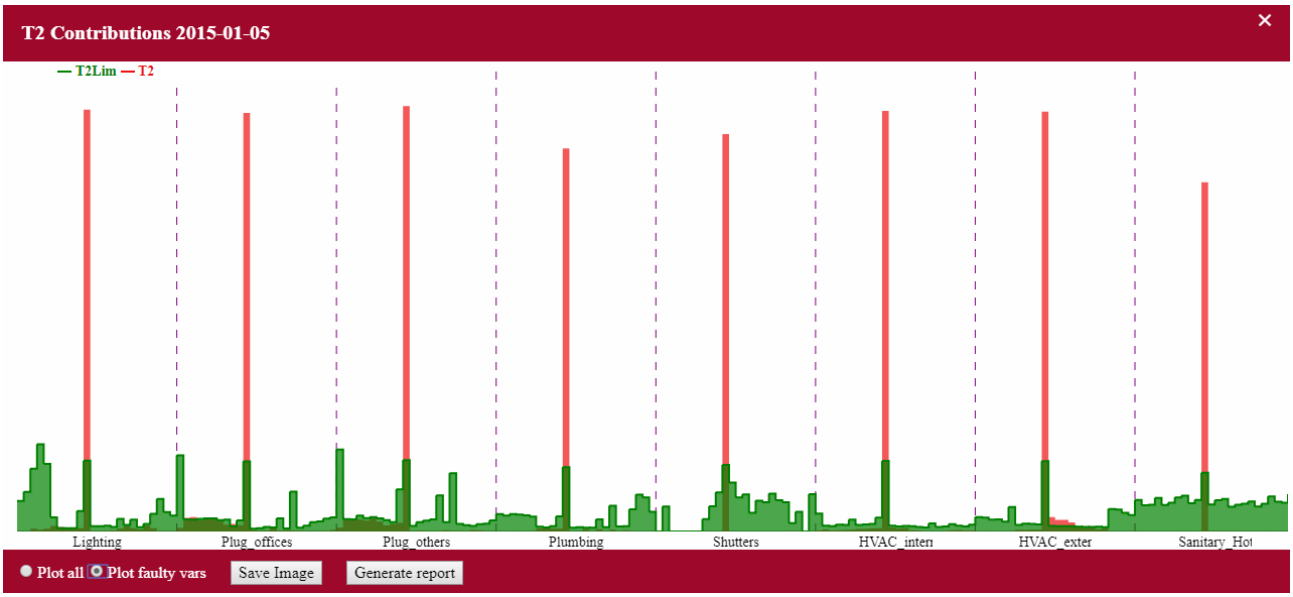



Figure 13 Contribution analysis showing only faulty variables

In addition, this new window has the button “Generate report”, which opens a new window in the browser with a different view of the faulty variables (Figure 14).

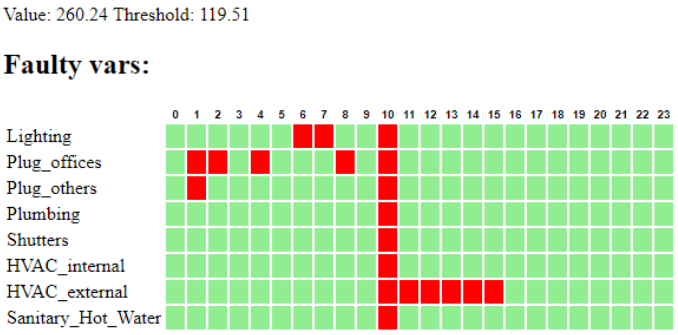



Figure 14 Detailed list of faulty variables

Optionally, and before clicking the “Calculate” or “Previous results” button, the option “Generate report” will show a new window in the browser with a complete report of faulty variables, if any (Figure 15).

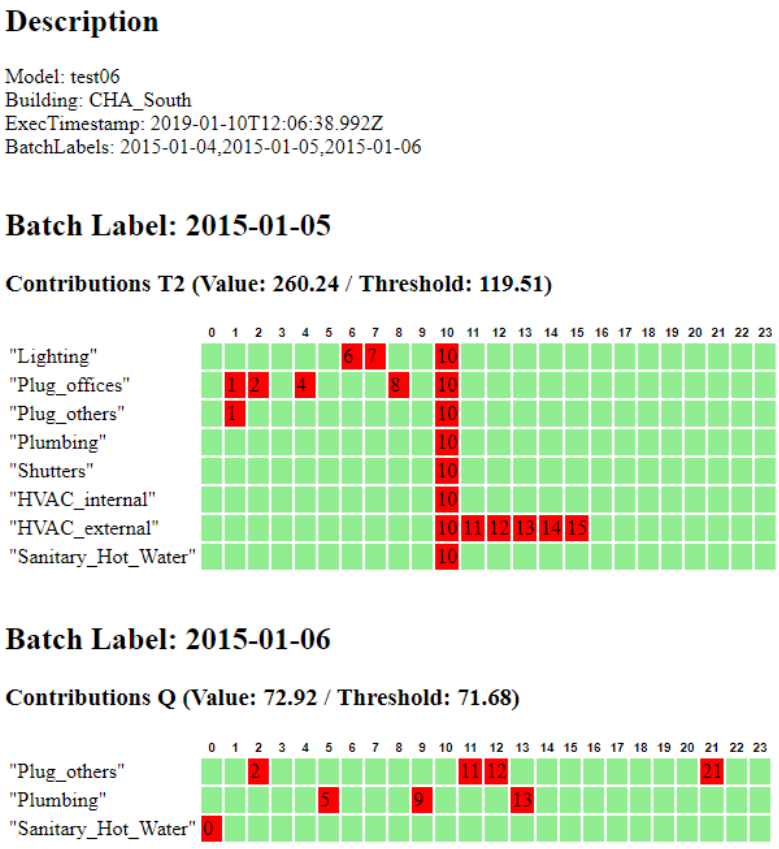



Figure 15 Complete report of faulty contributions

### 4.2.1 Previous results

This feature allows to graph results previously calculated for the selected model. So, first it is necessary to select a model, and the “Previous result” control will show all the available results (Figure 16). Click or double click over an option to visualize the charts.

NOTE: Using this feature it is also possible to generate reports such as described above.

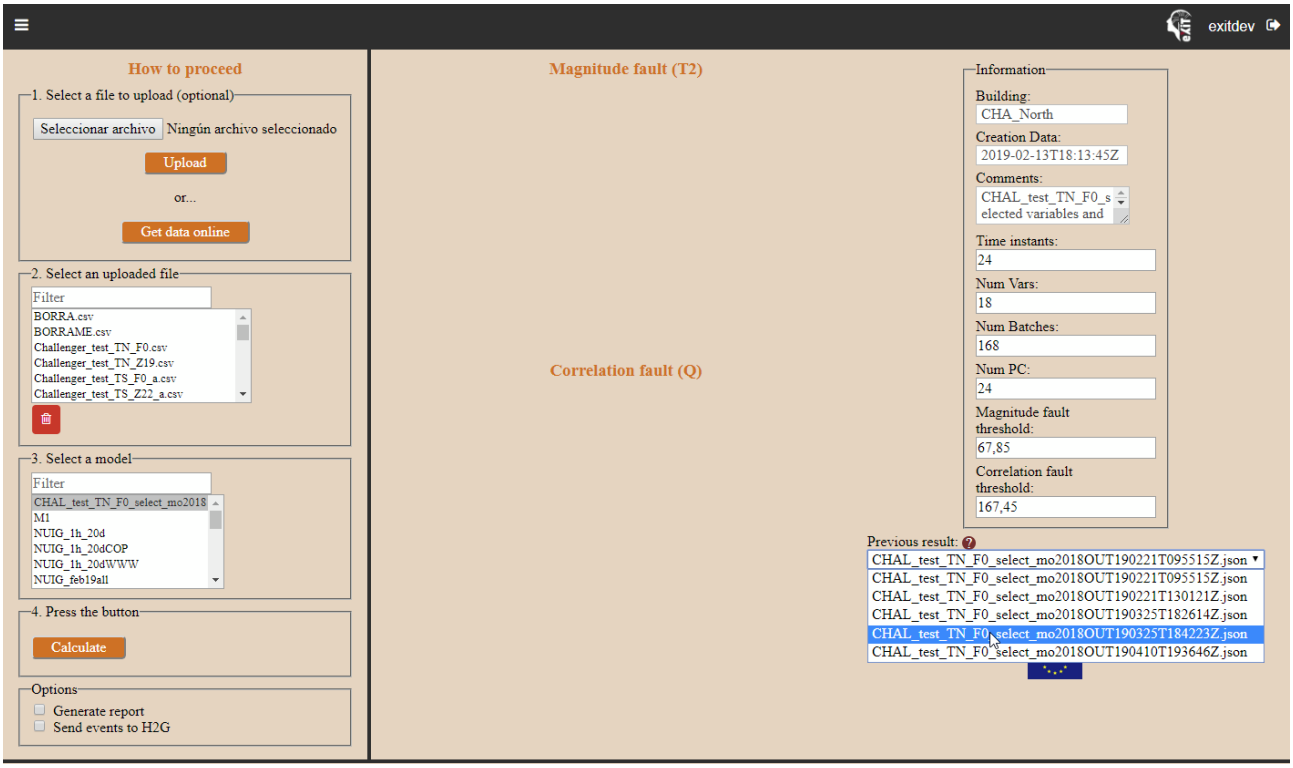



Figure 16 Selection of available previous results

### 4.2.2 Warnings

The FDD model is able to reconstruct input data if at least there is one value per batch. This feature enables the execution of the model, but we must be careful when evaluating the result. That is, if the present value is wrong, the rest of the values filled by the model probably will also be wrong. The user can recognise visually results obtained using filled data because the dots will be surrounded by a yellow circle. See an example in Figure 17. The warning will also be shown in the contributions chart (Figure 18) and the report (Figure 19).

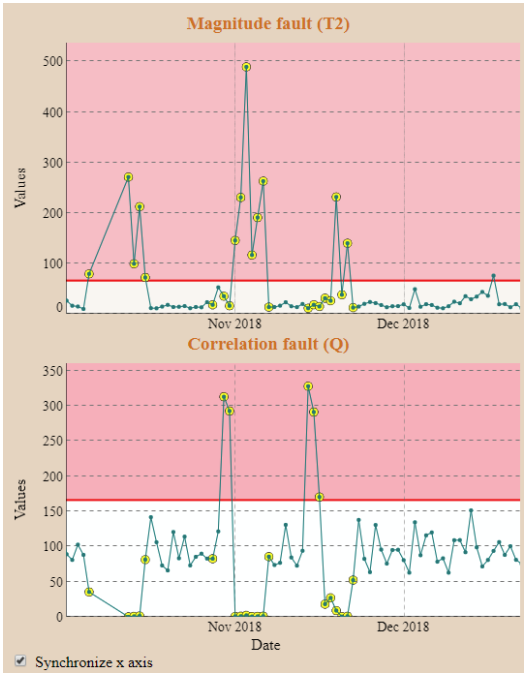



Figure 17 Visualization area detecting abnormal behaviour and warning (yellow circles)

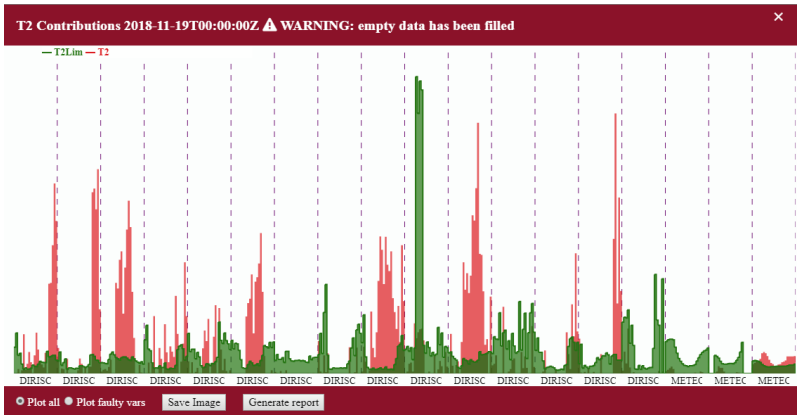



Figure 18 Contribution analysis of an observation with faulty variables that shows a warning

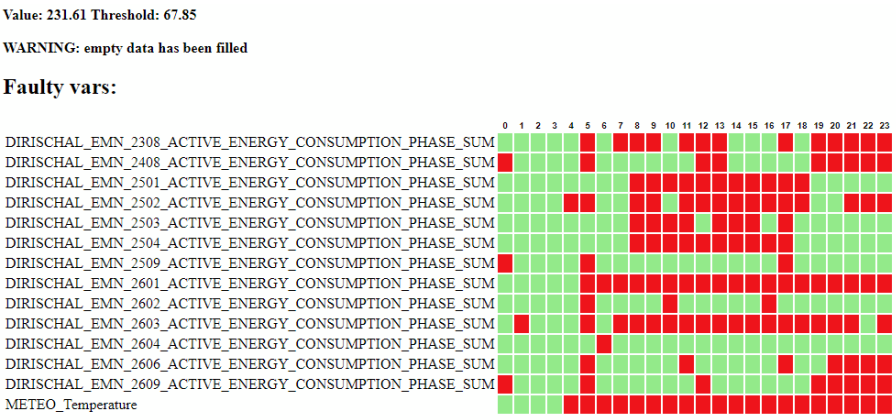



Figure 19 Complete report of faulty contributions that shows a warning

## 4.3 Scheduling of model executions.

In addition to the mode on demand, it is possible to schedule the executions of models (here called event). To do this, click on the menu button on the top left and select the option “Schedule events” (Figure 20). The modal window shown in Figure 21 will be displayed.

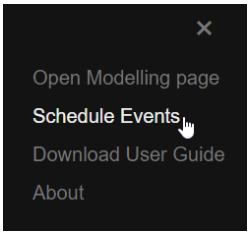



Figure 20 Options accessible from the menu button

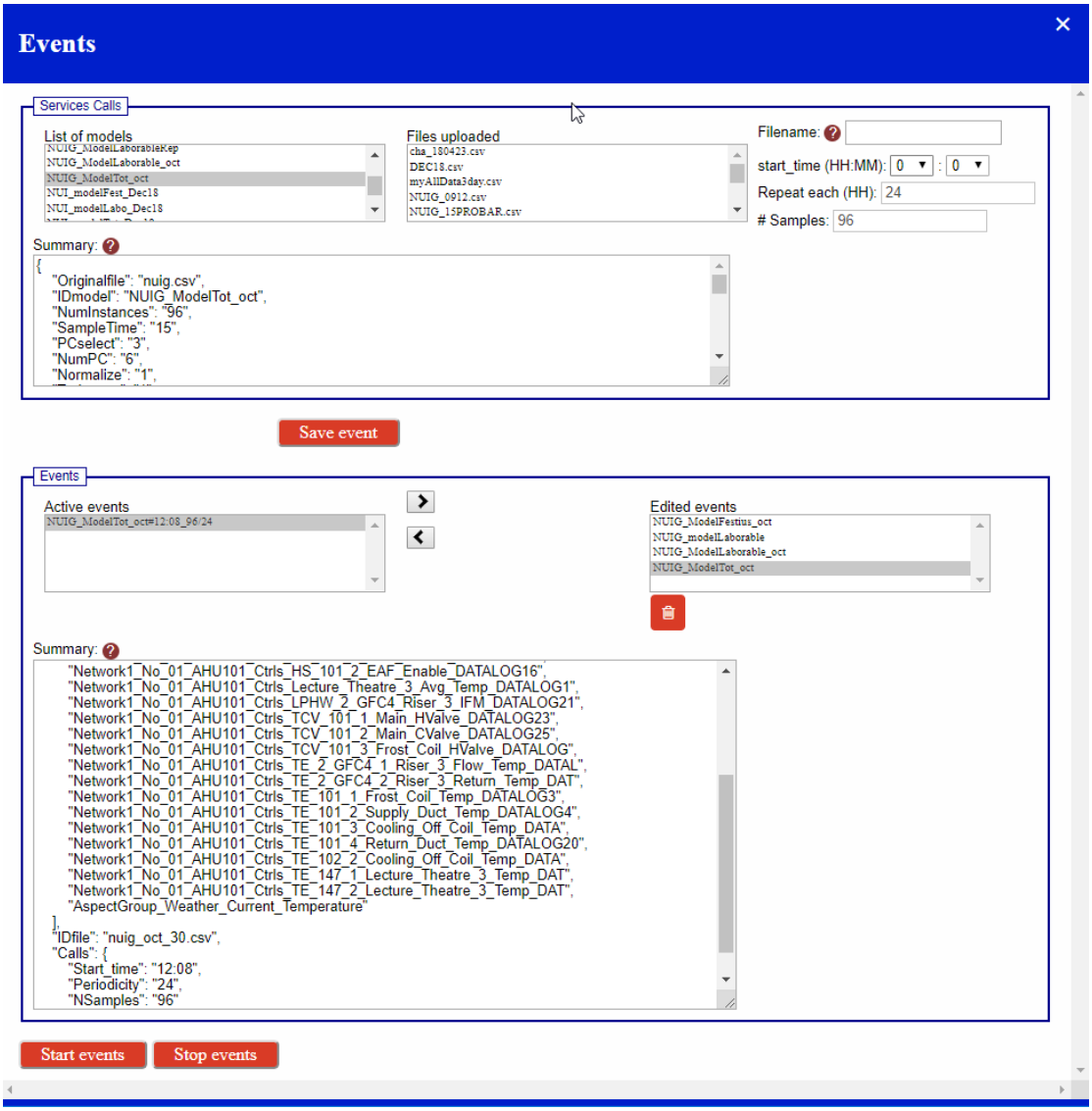



Figure 21 Configuration of scheduled events

The procedure to create and start a scheduled event is:

A. Create an event and save it.
   1- Select a model
   2- Select a CSV file, or introduce the name for a new file (without extension) if the file does not exist. Note that this file will be overwritten on each call.
   3- Introduce the start time to trigger the event, local time will be used. In order to get all the measures, it is recommended to start a few minutes past 2:00 a.m. or some hour later.
   4- Introduce the periodicity of the event in hours.
   5- Introduce (or check) the number of samples used in a batch for this model.
   6- Save the event. This new event will appear on the “Edited events” selector. The created events will be saved for a next session.

B. Activate the event.
   1. Choose an event on the “Edited events” selector. In the ‘Summary’ control will appear a summary of the event created. Modify it if necessary as described above.
   2. Click on the left button to export the selected event to the “Active events” selector. You can also remove active events from this selector with the right arrow button.

C. Start and stop events.
   1. Finally click on the “Start events” button. The active events will be saved for a next session.
   2. Started events remain active until the “Stop events” button is clicked and while the web client remains open.
   3. Currently, because the UdG server is able to execute automatically the active events, simply click on the “Start events” button and immediately click on the “Stop events”. This procedure it is enough to save the active events. The UdG server will recognize them and executes them. In this case the web client can be closed.

This mode uses the JSON file containing variables settings generated by the application (section 3.1.6), since during the automatic execution new data will be downloaded from the H2G core. This file was attached to the model when it was built. The schedule won’t run without this file.

NOTES:

- Only the active events will be triggered.
- When there are scheduled events running, the manual request is disabled.
- It is necessary to keep the application open in the browser so that the scheduled mode works. This not applies if the models are executed automatically by the UdG server.

# 5 Modelling interface

The modelling interface is shown in Figure 22. Visually, this interface is organized in two areas separated vertically. The left area correspond to the modelling parameters, and the right one corresponds to the management of the models.

## 5.1 Modelling Area

The steps to create a model are enumerated in the interface. Next, these steps are reviewed in more detail.

**1a. Upload a file**

This first step allows the user to upload a CSV file containing data to create the model.

Another option is to get data online (see section 3.1)

These steps are optional, if the file has been previously uploaded.

**1b. Select a file**

Here the user chooses one of the uploaded files to create a model, between the available files shown in the 'Files uploaded' control.

**2. Modelling parameters**

This section allows the user to introduce some parameters for modelling. These are:

***Time Instants***: number of samples of each batch. For example, for a daily model taking samples every hour, the time instants are 24.

***Sampling (min)***: sampling time in minutes. This is not necessary for modelling but for graphical and information purposes.

***PCselect***: method for selection of the number of principal components used by the model. There are 3 possibilities:

- **Kaiser-Guttman**: intuitively it consists of remaining all the principal components that provide at least as much information as one of the initial variables. This method tends to take less principal components than are really necessary.

- **Auto**: it is another automatic method that uses some graphical properties (internally) to select the number of principal components. This method tends to take more principal components than necessary.
- **Manual**: the number of principal components is specified in the “NumPC” box.

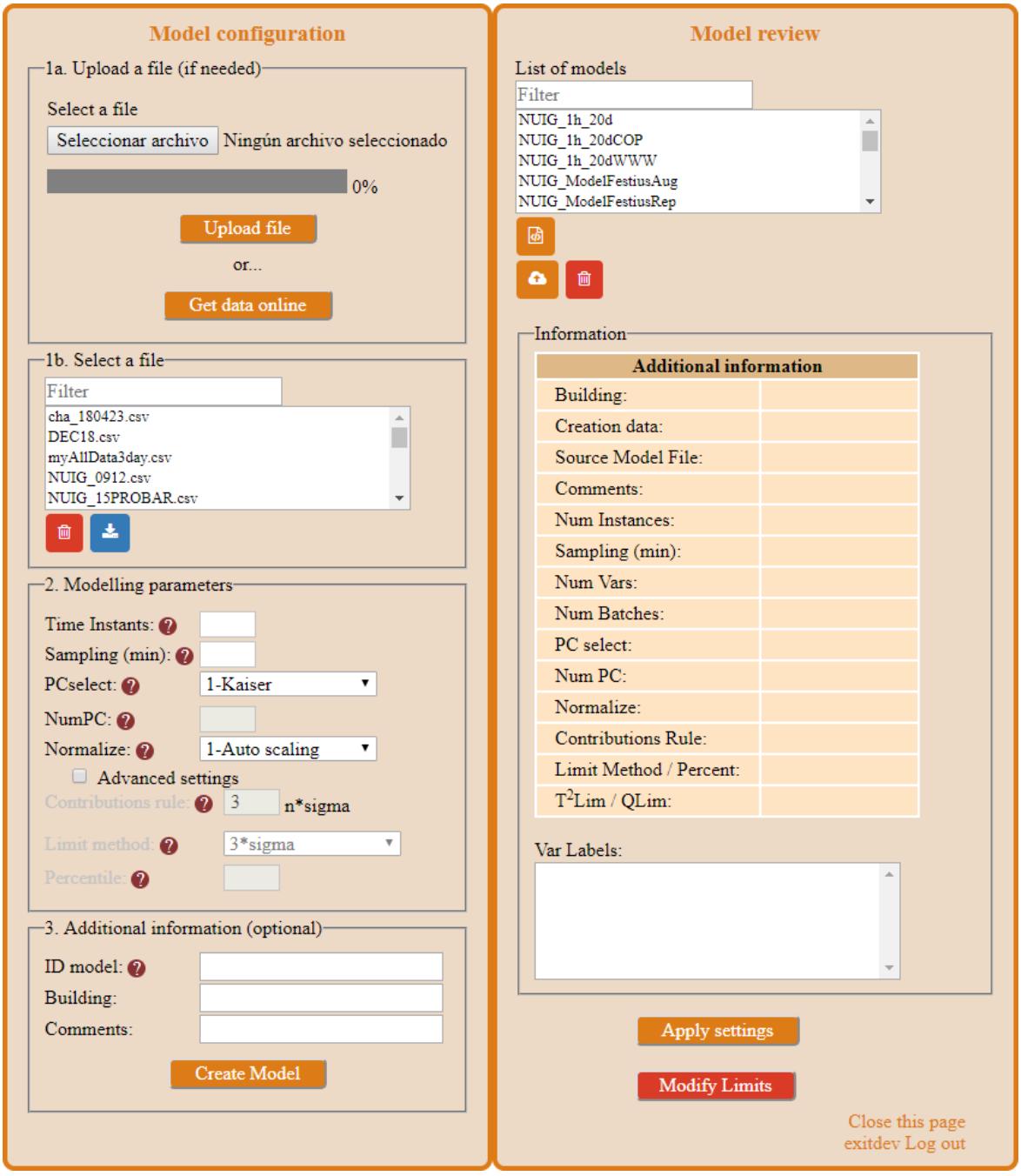



Figure 22 Modelling interface

***NumPC***: number of principal components used by the model, only available if the manual method was selected in PCselect.

**Normalize**: Normalization of variables. There are 3 possibilities:

- **Continuous scaling (CS)** – it assumes that the variables have the same distribution. A median and a standard deviation will be calculated for each original variable.
- **Group Scaling (GS)** – The median is calculated for each sample of each of the variables. The standard deviation is calculated for each original variable.
- **Auto Scaling (AS)** – Used when the variability changes throughout the process. The median and standard deviation is calculated for each sample of each of the variables.

**Advanced settings**: to be used by the administrator. It is not necessary for modelling.

#### 3. Additional information

This section allows the user to introduce some additional information, not necessary for modelling, but useful to remind the destination of the model. These are:

***IDmodel***: identifier of the model. If missing, it will be assigned by the web service. However, it is recommended that the user assign a name to recognize it easily. Valid identifiers must be written as character vector of alphanumerics (A–Z, a–z, 0–9) and underscores.

***Building***: here the user can introduce the building related with the model.

***Comments***: here the user can introduce other useful information or details about the construction of the model.

#### Creating the model

Once introduced all the parameters, click the “Create model” button. The modelling process lasts from a few seconds until several hours, depending of the chosen parameters.

**IMPORTANT**:

- Creation of a new model using an existent identifier will overwrite the old one.
- The server aborts modelling execution after 2 hours. Contact the administrator to obtain complex models.

## 5.2 Management Area

The right side of the interface shows the available models in the “List of models” control. Selecting one of them, it allows the user to view the settings and variables used by the model or delete it definitively. In addition, using the buttons under the list of models, the user could upload new models created offline.

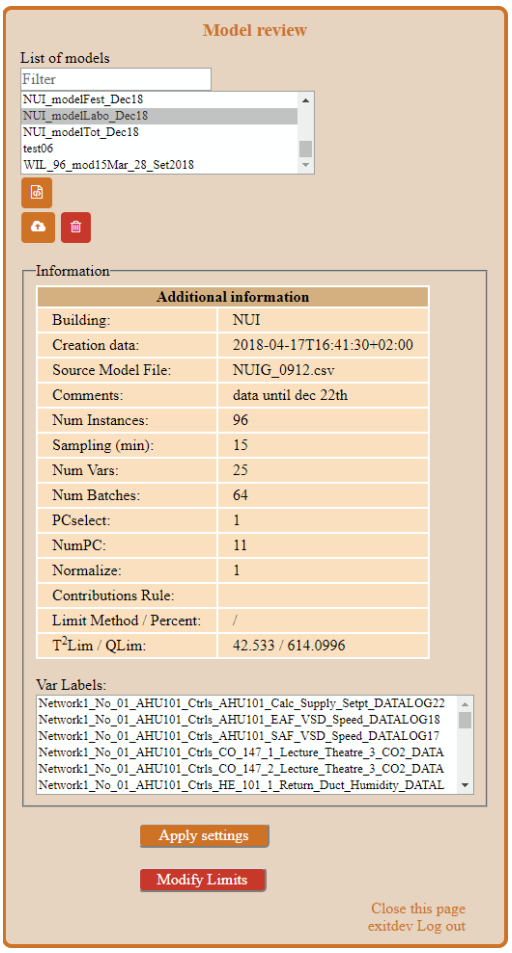



Figure 23 Management area in modelling interface

**"Apply settings" button**: when a model is selected it is possible to apply its settings to the left side of the interface to create a new model. This avoid the introduction of all parameters when only minor variations are necessary.

**"Modify Limits" button**: this button is used in conjunction with the advanced settings, to be used by the administrator.

# Annex A: data input format

The FDD algorithms have been implemented with the input data in CSV format, with the first row containing the identifiers of the variables and timestamps.

Regarding the format of this first row, it is mandatory:

- Identifier of timestamp must be: *TS*
- Other identifiers: a valid identifier is a character vector of alphanumerics (A–Z, a–z, 0–9) and underscores, such that the first character is a letter.

Regarding data, it is mandatory:

- Decimal separator: decimal point.
- Timestamp is a string using the ISO standard ISO-8601. It is UTC time.
- Column separator: semicolon.
- All the variables have the same sampling time and in each row all the samples have the same monitoring time.
- Every measurement must correspond to a disaggregated value, for example, the energy consumed every hour.

## *Training Data*

Data have to be according to operating conditions that want to be modelled, usually NOC (normal operation conditions).

Data set needs to be real data (No reconstructions), and batches must be complete and without blanks. If data has blanks (NaN values) all the batch will be discarded.

## *Exploitation Data*

In this case, the module does not need real data (reconstructions allowed) also missing values are allowed. If there are missing data for some timestamp, it needs that the record for that timestamp exists but without any value inside. Observations with missing data will be reconstructed.

## *Example:*

TS;AHU101_CO_147_1_3_CO2_DATALOG10;AHU101_CO_147_2 _3_CO2_DATALOG7

2017-01-01T00:00:00Z;422.798278809;416.026885986

2017-01-01T00:01:00Z;422.334899902;414.005615234

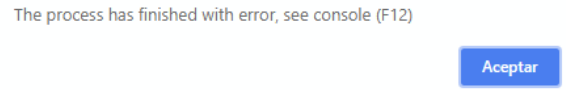
2017-01-01T00:02:00Z;422.206390381;412.897644043

# Annex B: Errors

**Error**: Button “Schedule events” remains disabled after entering to the monitoring page

**Cause**: the web service is not responding.

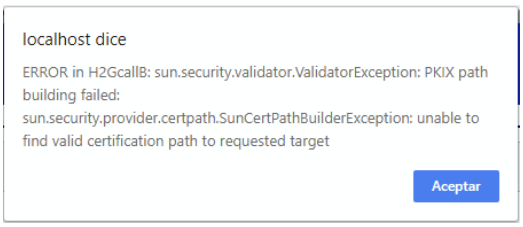
**Error**: Alert building a model.




Console: Error in postModelling: ERROR: DATA/setRawDataVW: data must be double or single type.

**Cause**: Bad format of the input data file.

**Error:** ERROR in H2GcallB: sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target




**Cause**: The data provider has changed the SSL certificate (java).

# Annex C: Tips for exploring results

The common execution periodicity in the models is 1 day. That is, once a day the module analyses the data of the day before, which offers a single point on the control charts. The module notifies users through email in two situations:

- When the model could not be executed due to lack of data. The module is configured to not accept inputs with less than 50% of data. In itself this is already an alert that something abnormal happens with the storage or transmission of data. Ideally, the pilot's user should check what's wrong with it.
- When the model detects an abnormal state. Ideally, in these cases, the pilot's user should check if the alert is relevant.

Since daily monitoring is not frequent, a subsequent analysis of many days can be tedious, since the results are saved for single days. Next some tips to explore data are described.

**Download again the data:**

This is a simple process. Click on the "Get data online" and open the "Files" tab (Figure 24).

Select the model in the "List of models" selector. The full configuration for this model will be shown in the "Summary" control.

Click on the "to Fine-grained" button. The needed variables for this model will be shown in the "Fine-grained selection" table. Select all.

Write a name for the CSV file and click on the "Get CSV" button. Once the process is finished, close this window.

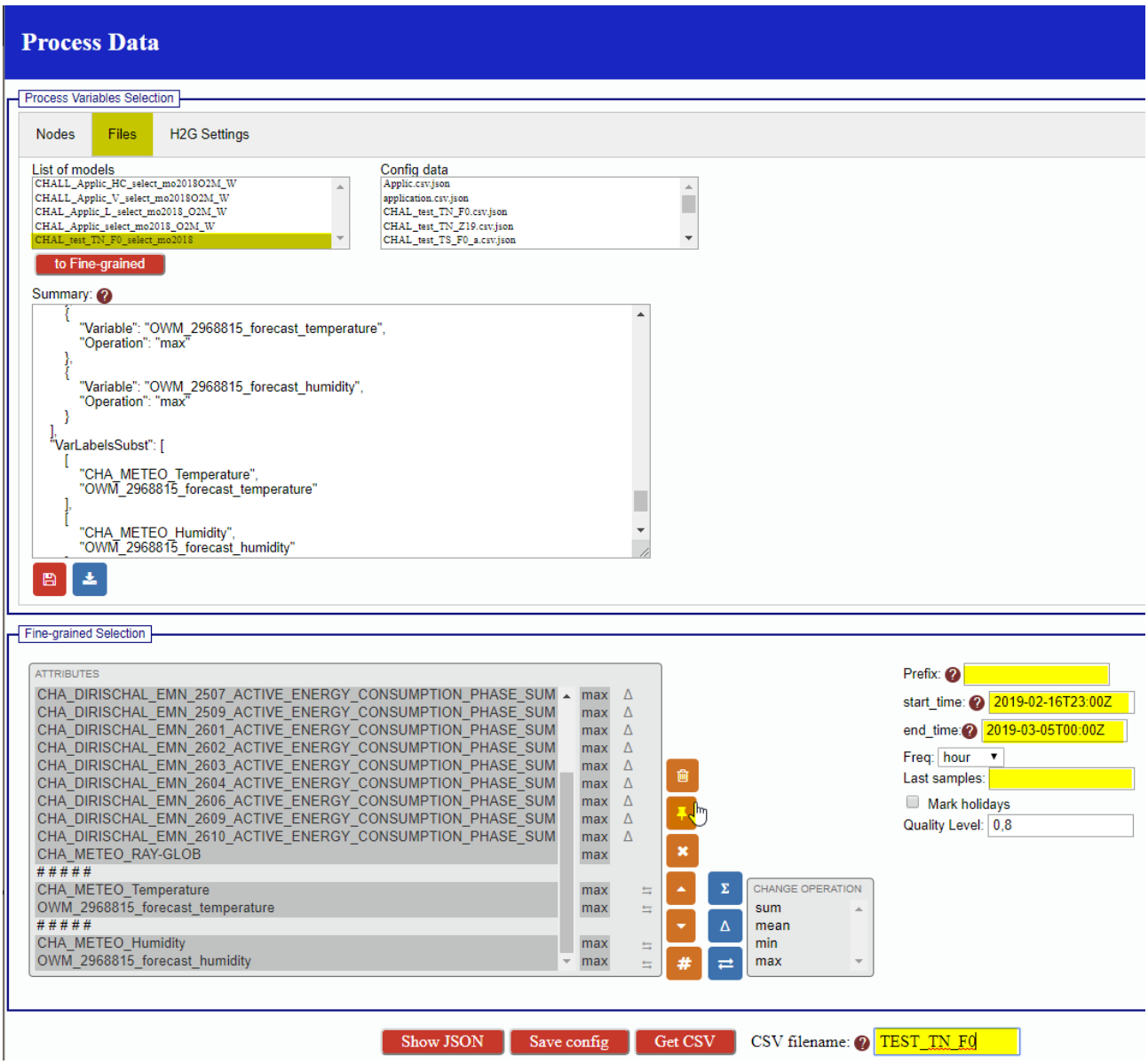



Figure 24 Getting data based on a model

**Project new data into the model:**

Back to the monitoring window, select the downloaded CSV file and the model (Figure 25).

Click on the “Calculate” button with the “Generate report” option activated. A new window with the complete report of faulty contributions will be shown (Figure 26), in addition to the updated control charts (Figure 25).

**NOTES:**

- The “Generate report” is optional. The user always can see the faulty contributions report if checks this option before loading the available results in the “Previous result” list. Once the results are obtained it is not necessary to calculate again.

- Make sure the “Prefix” field is empty, since when the variables are extracted from the model, they already contain the prefix.
- If at least one variable use the delta operation, it will be needed to download one sample before the starting time. See the figure as example, in this case the frequency is 1h.
- Make sure the “Last samples” field is empty.

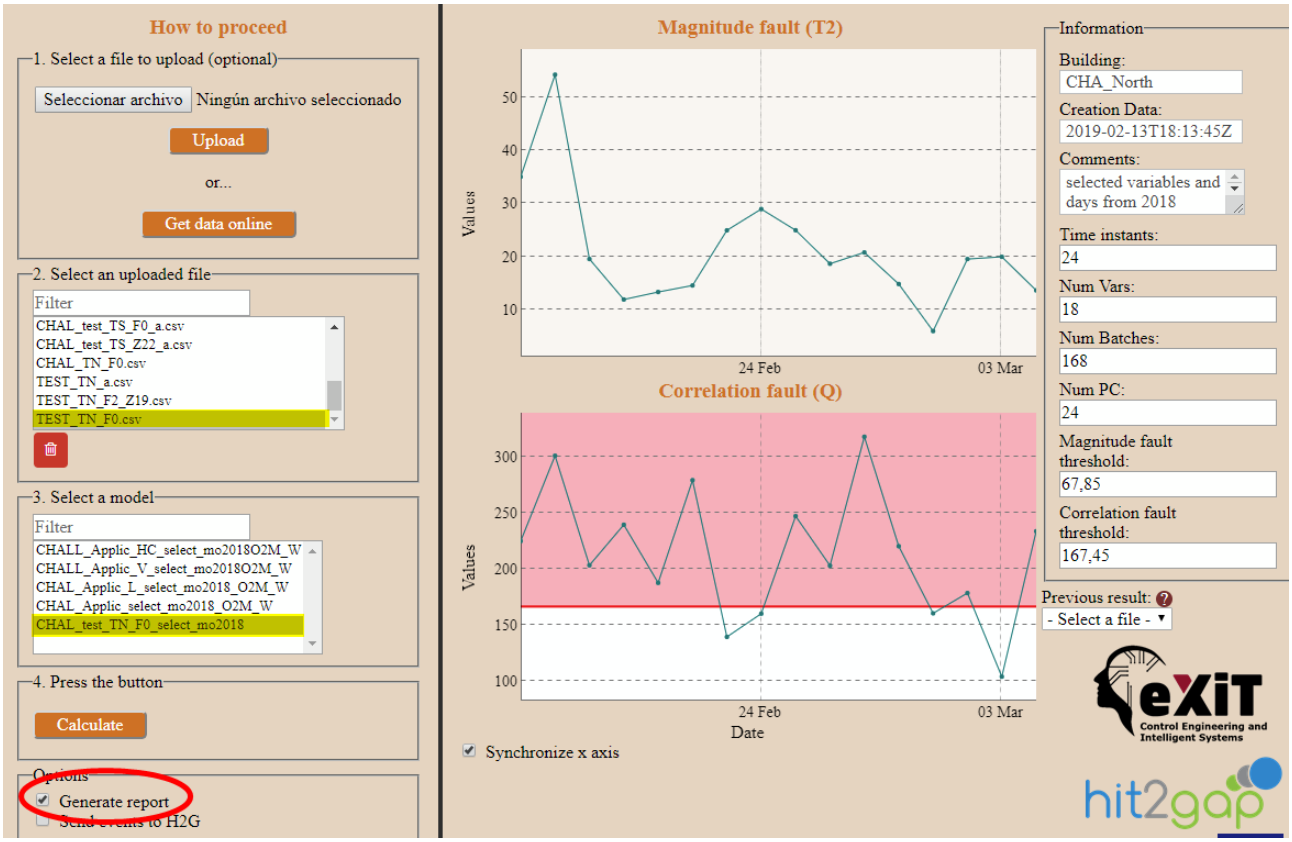



Figure 25 Monitoring window

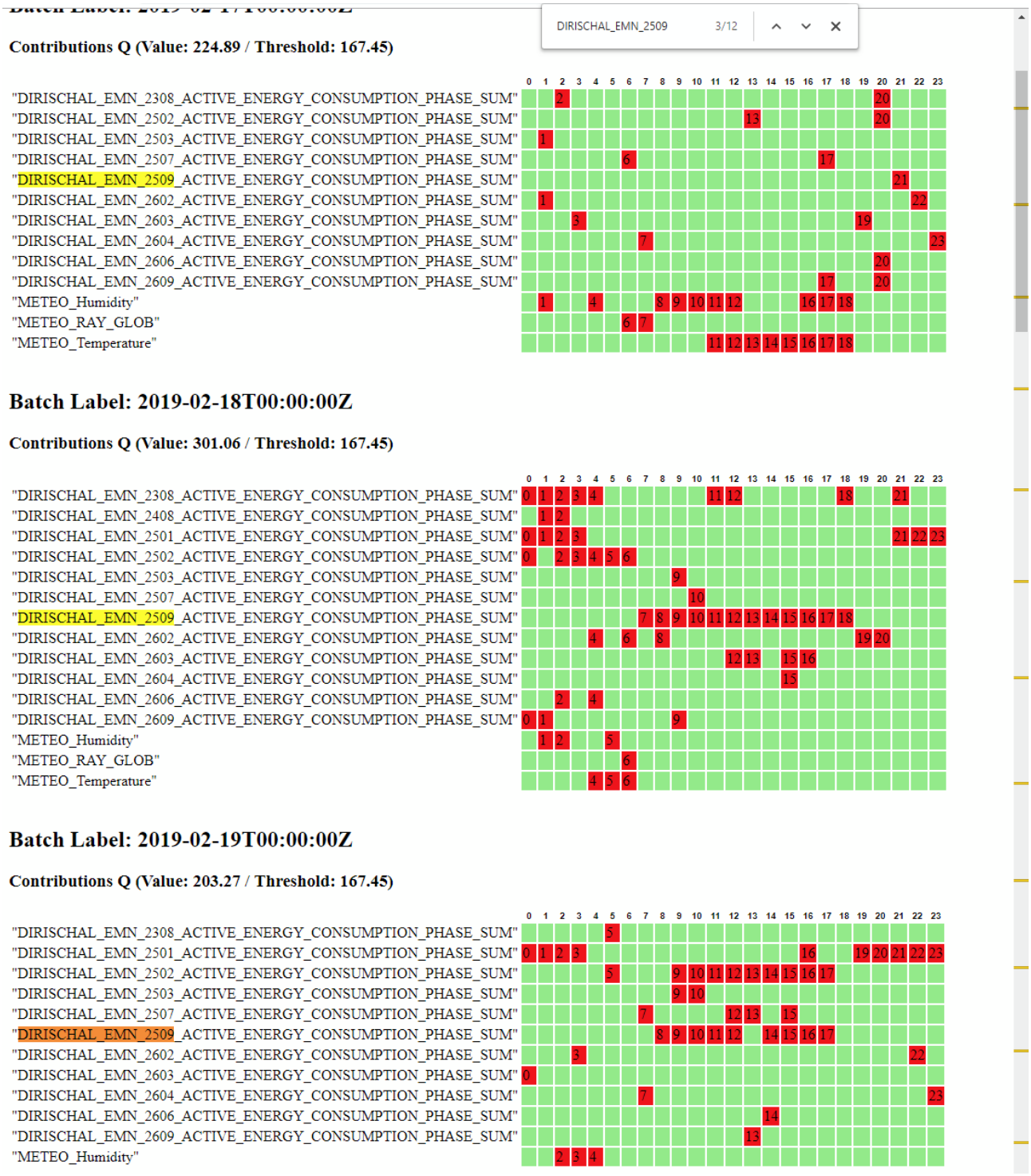



Figure 26 Faulty contributions report

### Visualizing the variables:

Finally, once the new CSV is created it is possible to download it to visualize the data in any program that accepts this format (Figure 27).

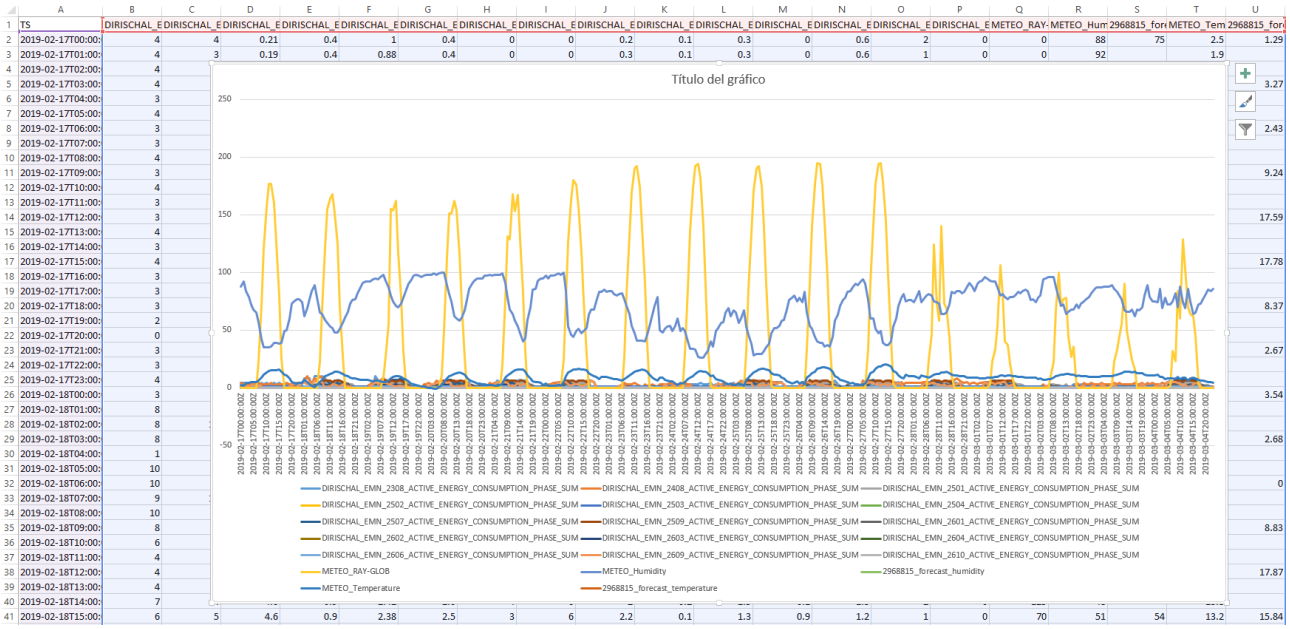



Figure 27 Example of data visualization using Excel.