SOLDIER: SOLution for Dam behaviour Interpretation and safety Evaluation with boosted Regression trees

User Manual: Version 4.0

Contents

1. Introduction

1.1 Purpose of the software

The main goal of this application developed by CIMNE is the analysis of dam monitoring data through the interpretation of relationship between variables using machine learning (ML) models.

As the entire observation process supported by this software makes use of regression analysis, it is assumed that a user is familiar with the basic concepts of this topic.

1.2 Basic Concepts

This app requires a .CSV, .XLSX or .RDS[1] file to be uploaded. SOLDIER has been developed for the analysis of the influence of some inputs on a target output. This analysis is done by applying a machine learning predictive model on a set of input data. The reliability of the model interpretation is related to its accuracy when predicting a response. In this sense, the application allows to modify the parameters of the model as desired and to select a certain time period to remove from the training set and use only to test the accuracy of the model. Fitted models can be stored and later used in R.

1.3 Main work flow

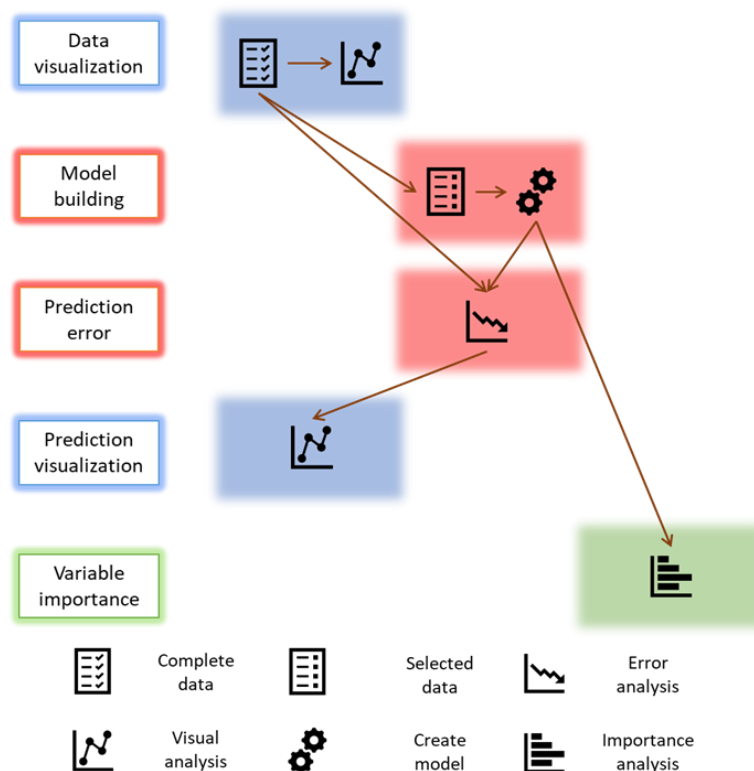The five steps typically followed for data analysis in SOLDIER are shown in Figure 1.

Figure 1.   Workflow.

---

[1] .RDS is a specific format for R language.

1.4 User interface

The user interface is split up into a menu area on the left and a working area on the right. The working area features three tabs that can be accessed from the left-column menu. The first one, "Data exploration", includes options to load data into the app and visualization tools with interactive functionalities. Different types of plots are available to facilitate the analysis process.
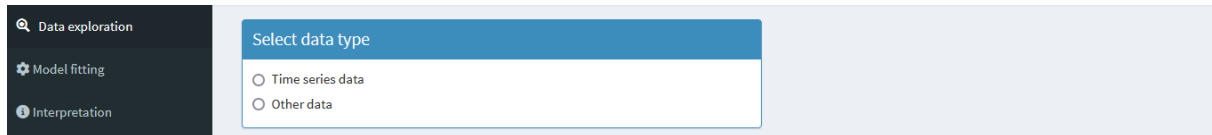

Figure 2.   Sidebar menu and initial window of the application.

The second tab, "Model fitting", allows the user to fit a predictive model for a given target variable. The user can select the target variable (response to be predicted), as well as the variables to be considered as predictors (inputs). The algorithm used is "Boosted Regression Trees" (BRT) [2], which has proven to be advantageous in previous works [3].

The interpretation of the model and the relationship between variables is done in the third tab, which features options to interpret the predictive model previously fitted: both the strength and the shape of the association between the considered inputs and the response are computed and visualised. For this purpose, the application makes use of two interpretation tools:

- Classification of predictors in terms of the strength of their association with the target variable under consideration. The relative influence of each variable is shown in the form of histograms.
- Partial dependence plots show the average effect of each input on the response, as learned by the model.

All these plots can be included in safety reports or in any other text document. Likewise, the model can be saved once fitted for further analysis. In the following sections, the included functionalities are described in detail.

*Note: The interface has been designed for a minimum resolution of 1440*900 (lower resolutions would result on wrongly fitted boxes).*

2.   How to start the application

The application was coded to run locally. It is based on R [1], an open-source software environment for statistical computing and graphics that can be compiled and run on Windows, Mac OS X, and numerous UNIX platforms (such as Linux). Once installed, the application runs as a standard RStudio Shiny [4] application.

RStudio is a separate open-source project that brings many powerful coding tools together into an intuitive, easy-to-learn interface that requires low system requirements. The RStudio program can be run on the desktop or through a web browser. This desktop version is available for Windows, Mac OS X, and Linux platforms (Figure 3).
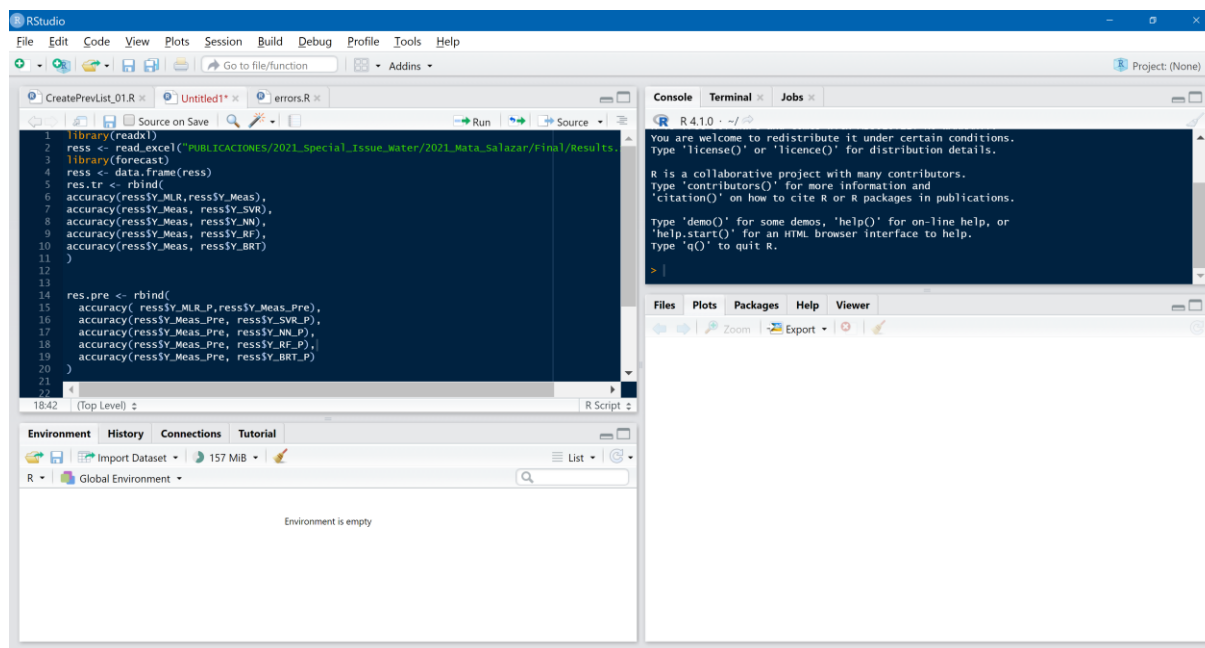
Figure 3.   User interface of RStudio.

The application runs like any other R-shiny application. These are the recommended steps to follow:

1. Copy the files on a local folder:
   a. R files: ui.R, server.R, global.R, aux_soldier.R
   b. text files: AUTHORS.txt, LICENSE, README.md
   c. folders (and the files inside): manual, tests, www
2. R version 4.3.1. or later needs to be installed in the system. For most platforms R is distributed in binary format for ease of installation (https://cran.rstudio.com/).
3. In case Rstudio is not installed, it can be downloaded from: https://www.rstudio.com/products/rstudio/.
4. Open ui.R file using RStudio (File/Open File or Ctrl + O).
5. Press button "Run App". The first time you run it the software will install the necessary packages (that can take a long time) and maybe you should press "Run App" again. (Figure 4).
6. The application opens in the default web browser window

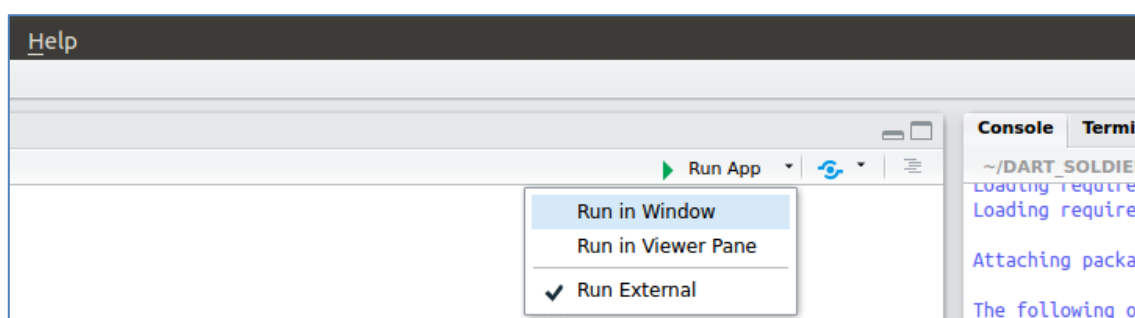*Note: The three bars on the right upper side can be used to hide the side menus on the left*

Figure 4.   "Run App" button appears once some of the .R files has been opened in RStudio.

3. Quick Go-Trough

The objective of this section is to give the user an idea of the application by covering the basic steps necessary to set up a model that can be used to analyse dam behaviour.

3.1 Initial data

The first required step is to load a database through the first tab. Select "Time series data" from the menu (Figure 5). A window opens up to browse through the folders on the computer allowing to choose the desired data file.

Figure 5.   Options for loading data.

3.2 Model building

In the second tab (Model fitting) the predictive model is created, for which the target variable needs to be selected (Figure 6), as well as the inputs (box 1). Then, the default time period for the analysis can be modified. In general, part of the data is used to create the model and a different time period is considered to verify its accuracy (box 2).

Once the calculation ends (box 3), the accuracy of the ML model can be checked through error measurements (box 4) or by comparing the model prediction with the actual data (box 5). If the errors are considered to be inadequate, a new model can be created with different parameters.
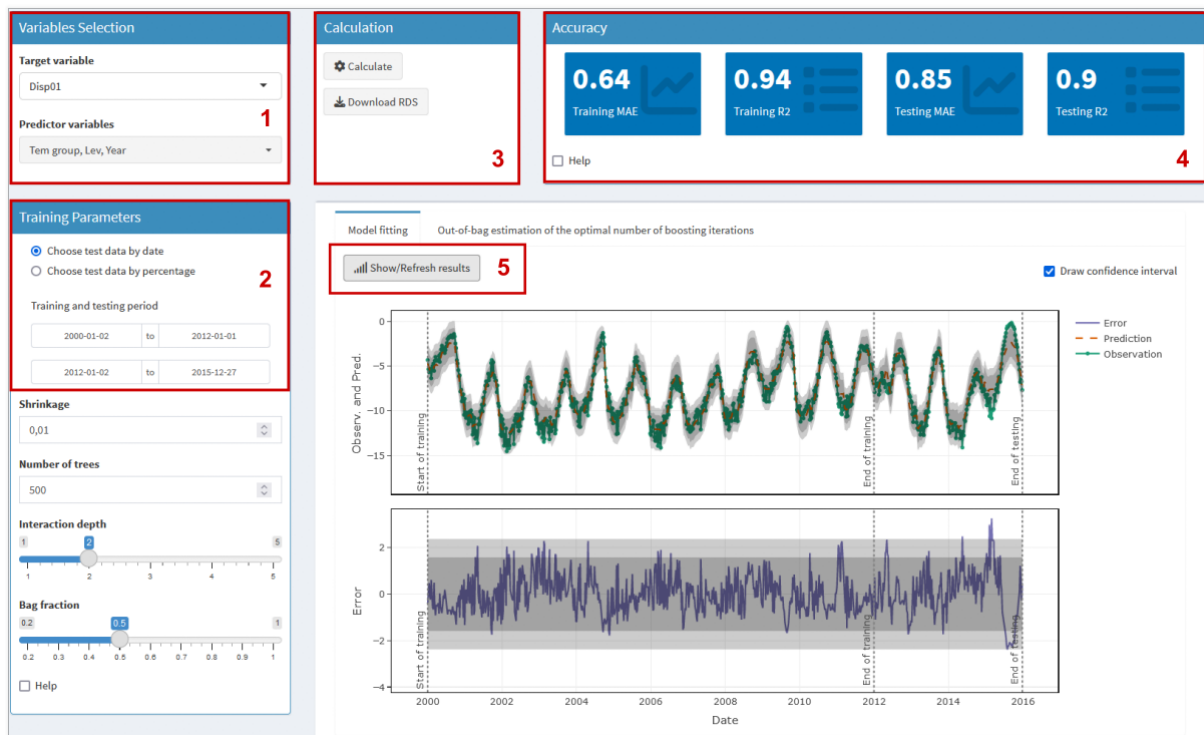

Figure 6.   Interface for "Model fitting" tab.

3.3 Importance analysis

In the last tab (Interpretation) the influence of the inputs on the target variable can be studied either individually (Figure 7 left) or combined (Figure 7 right).
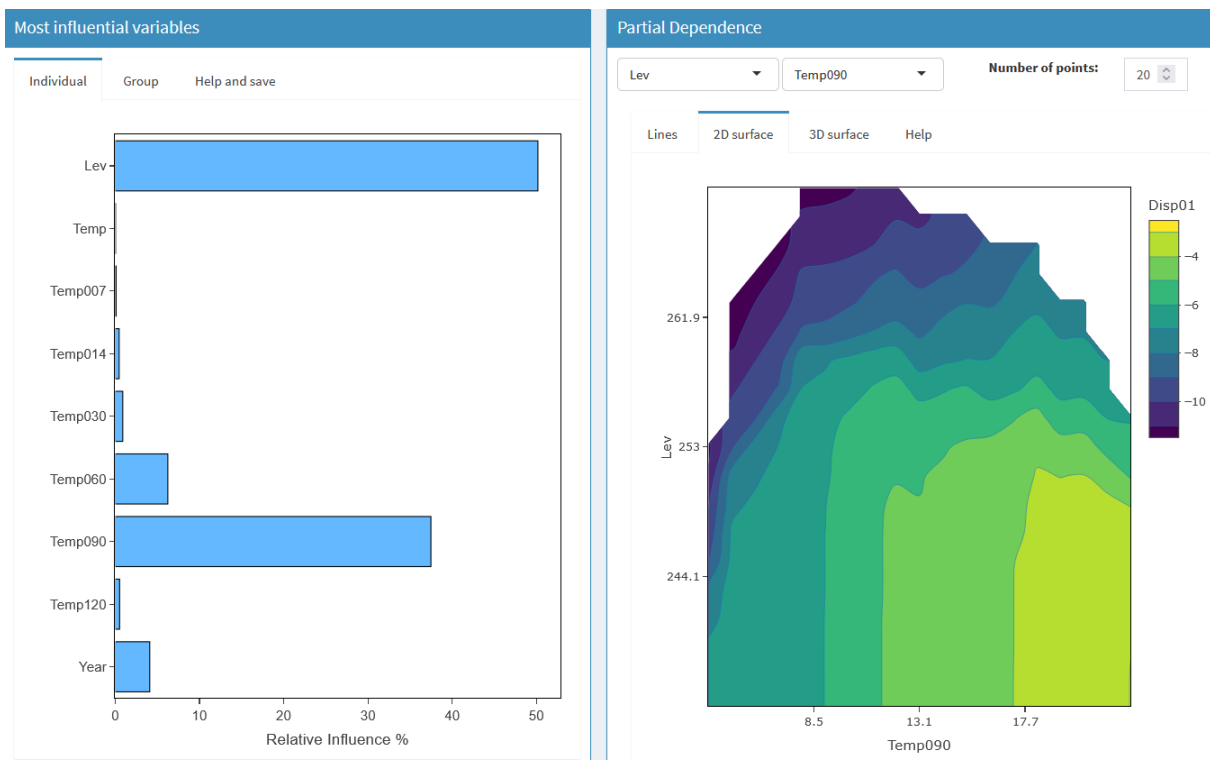

Figure 7.   Model interpretation.

4.   TAB 1: Data exploration

The first three items in the column on the left are tab menus. Their options are shown when clicked. The first of these items is "Data exploration".

4.1   Data options

The user can load a new data file to create new prediction models. The application allows the user to search for the appropriate files through the folders on the computer. The user can create an ad-hoc .CSV, .XSLX or .RDS file up to a maximum of 50 Mb to be loaded and analysed. If an Excel file is loaded, all the data must be on the first tab. If the "Time series data" option is selected (Figure 5), the first column must include the time/date in "day/month/year" format[2]. The first row must be a header with the names of the variables, which must meet certain limitations:

- Names shall not contain empty spaces.
- The first character must be a letter (ñ and ç included) or one of these symbols: { } . : º

---

[2] This manual refers to the "Time series data" option. Everything applies as well to the "Other data" option, except for the Predictions vs observations plot, which is shown as scatterplot instead of as time-series.

The user can choose among three types of exploratory plots in order to visualize the loaded data in the first tab of the central box. If a model was already fitted, the residuals (difference between predictions and observations) can also be plotted.

Figure 8.   Options for plotting data.

The second tab of the central box allows loading an image of the structure to help understanding the names of the variables and the location of the devices.
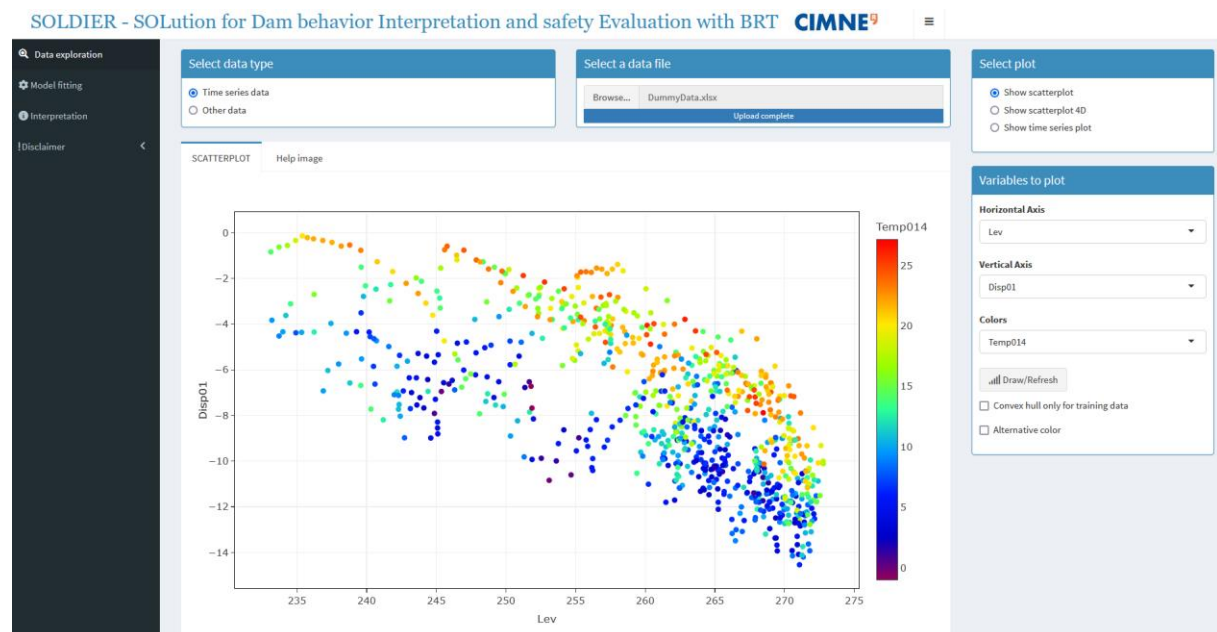

Figure 9.   Data exploration tab with an example scatterplot.

4.2   Plot options

The first option is a scatterplot where relationship between variables, as well as their evolution over time, can be observed. Inputs can be analyzed by plotting them against other variable to get a visual idea of how they are related. The user can select which variables to display on each axis, as well as a third option that allows the colour of points to be adjusted based on some of the variables in the data set.

Figure 10.   Drop down menus for selecting variables to be displayed in the scatterplot.

The plots are generated with the plotly library, which provides interactivity. For instance, the values of a given point data can be shown by hovering (Figure 11).
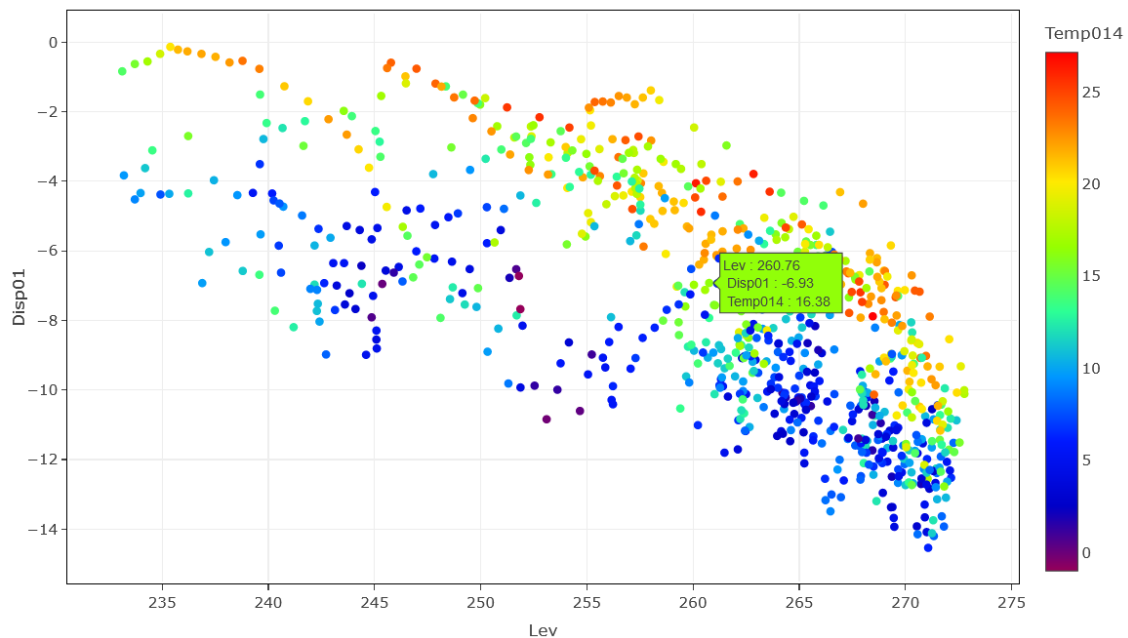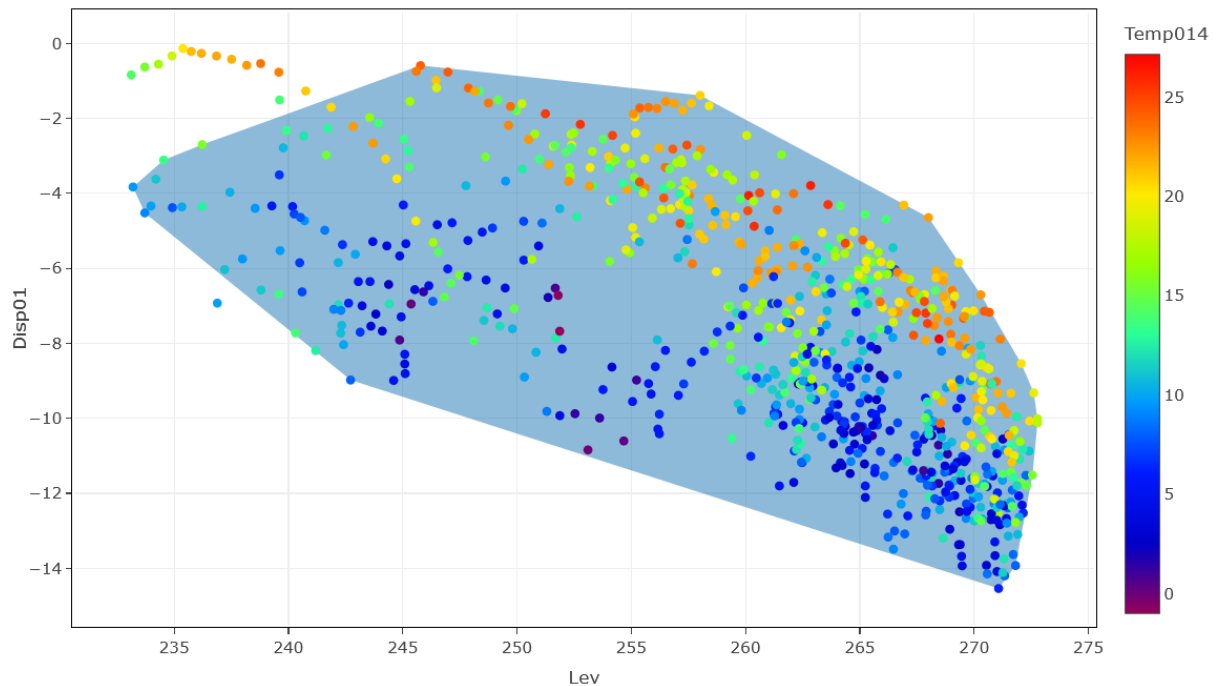

Figure 11.   Values are shown by hovering over the scatterplot.

Other interactivity options include zooming or exporting as png file.

Once a predictive model has been fitted, this plot allows for drawing a polygon enclosing all the training data (convex hull). This allows for identify data in the test set out of the range of the training set.

Figure 12. Scatterplot with convex hull. In this case, some data in the testing set are out of the range of the variation of the inputs in the training set.

Some combinations of colours may be difficult to visualize. To overcome this problem an alternative colour scale and background is available. It can be activated marking the "Alternative colour" box.

The data can also be shown as a scatterplot in 3D (Figure 13). The same library was used and the options are basically the same, except for the convex hull.
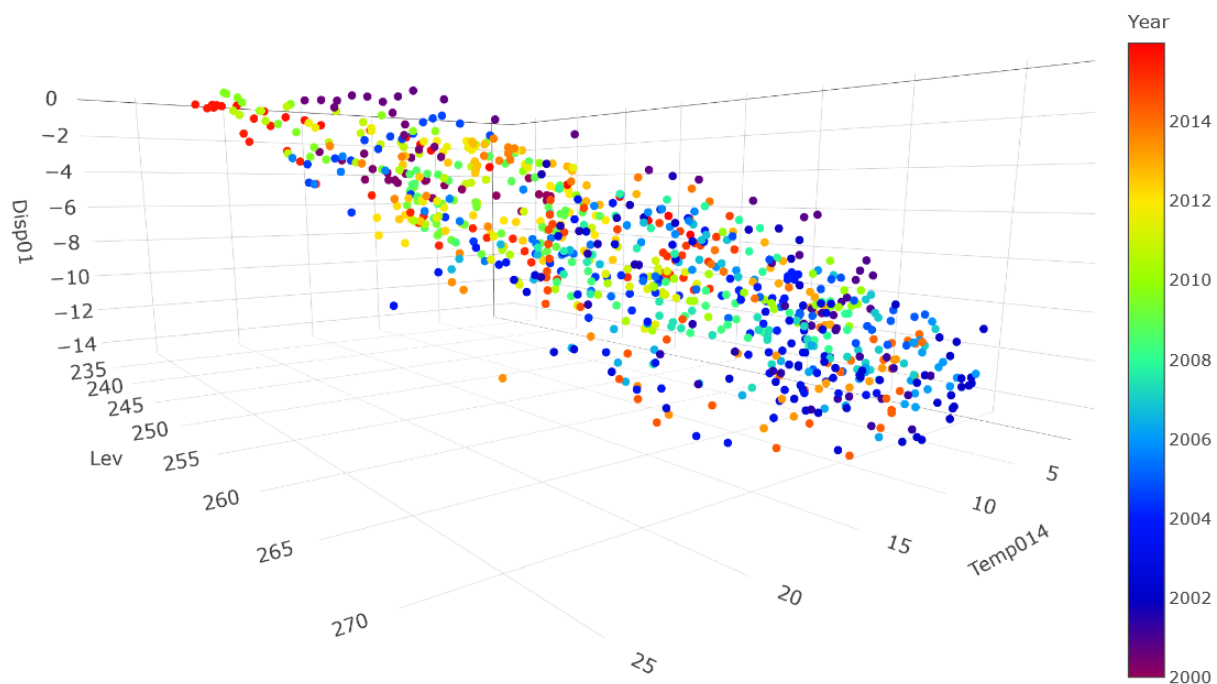
Figure 13.   Scatterplot 4D (three axes plus colours).

Finally, the data can also be shown as time series. A secondary axis can be used on the right to facilitate the display of variables with different range of variation, as shown in Figure 14.
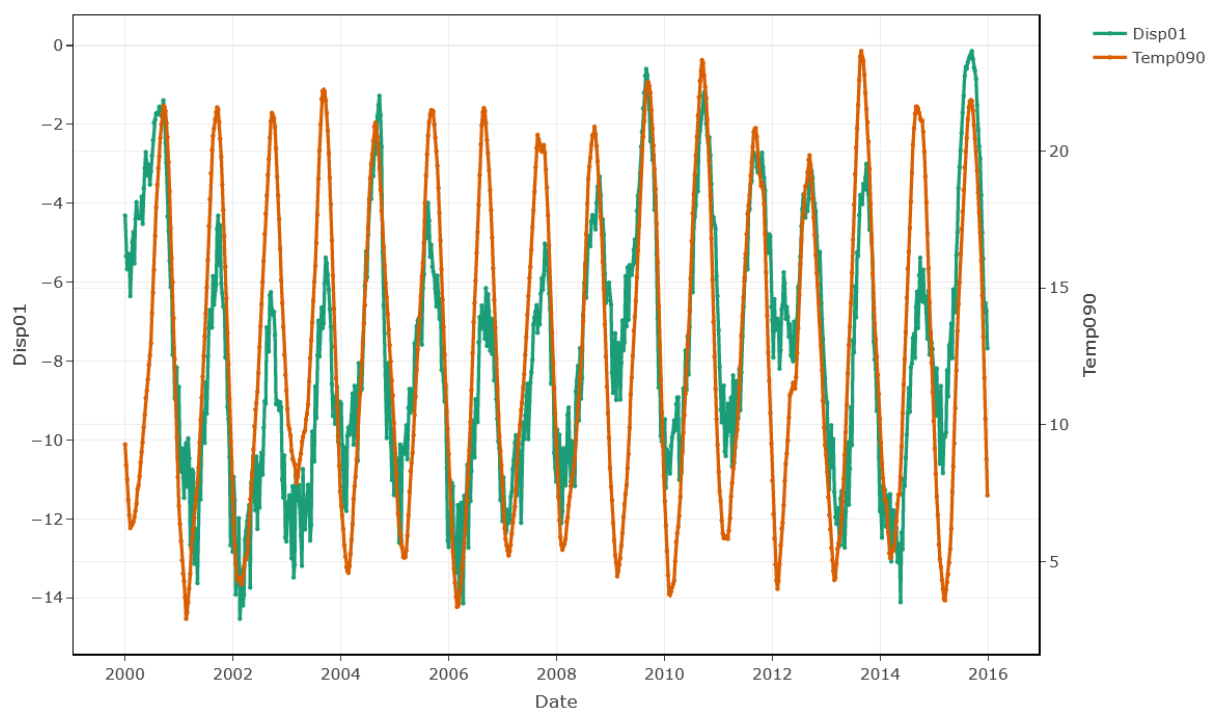

Figure 14.   Time series plot.

5. TAB 2: Model fitting

In this tab, a BRT model is analysed or fitted with the selected data and parameters. The interface features 5 boxes described herein.

5.1 Selecting target and predictors

The variable to be predicted needs to be chosen in the "Target variable" menu. Also, the predictors can be selected in the "Predictor variables" menu, which allows for multiple selection (variables starting with the same 3 letters can be selected as one group).
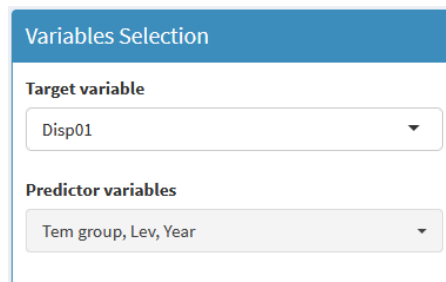
Figure 15. Box for variables selection

Note: The target variable must be numerical and not integer.

Note: Dates cannot be used as predictor. In the test dataset, a variable "Year" is included, with the date as numeric, to account for the time effect.

5.2 Choosing model parameters

The training parameters of the model can be defined with the controls in the lower left box. They can be grouped into two classes. First, there is the selection of the training and test periods. By default, training period is set as the initial 75% of the available time while the most recent 25% is reserved for test. The user can modify them by entering the start and end dates of each period, or the percentage of data for testing.

Note: If the train and test periods are selected by date, the testing period should always start just after the end of the training period.

Figure 16.   Options for model parameters.

According to previous results, the models based on BRTs are relatively robust, and therefore their results vary little with the modification of the training parameters. In most cases, reliable results are obtained with the default values. However, it is advisable to perform some tests with different values. The parameters that determine the training process are:

- Number of trees
- Interaction depth (number of splits it has to perform on a tree)
- Bag fraction (proportion of data to be selected at each step)
- Shrinkage (learning rate of the trees)

"Number of trees" indicate the number of iterations for the algorithm while "Interaction depth" controls the complexity of the trees in the ensemble, i.e., the number of branches. "Bag fraction" is the proportion of training data used to fit each individual tree. This prevents overfitting. Finally, "Shrinkage" is applied to each tree in the ensemble to prevent overfitting by reducing their effect in the final prediction. Low values are better for that purpose, though more trees will be needed for equal predictive performance.

5.3   Building new models

Once the data and parameters are set, the model is fitted after clicking on the "Calculate" button in the "Build models" box. Then, a model is fitted and the results are shown.

Note: If there are missing date in the target, those rows will be excluded from the calculations (including target and inputs).

Along with the progress of fitting models, different boxes show messages with information about the training process, including warnings (for example, if the range of any input for test data exceeds the one for training data), if appropriate. Once the computation ends, the resulting model can be saved into an .RDS file.
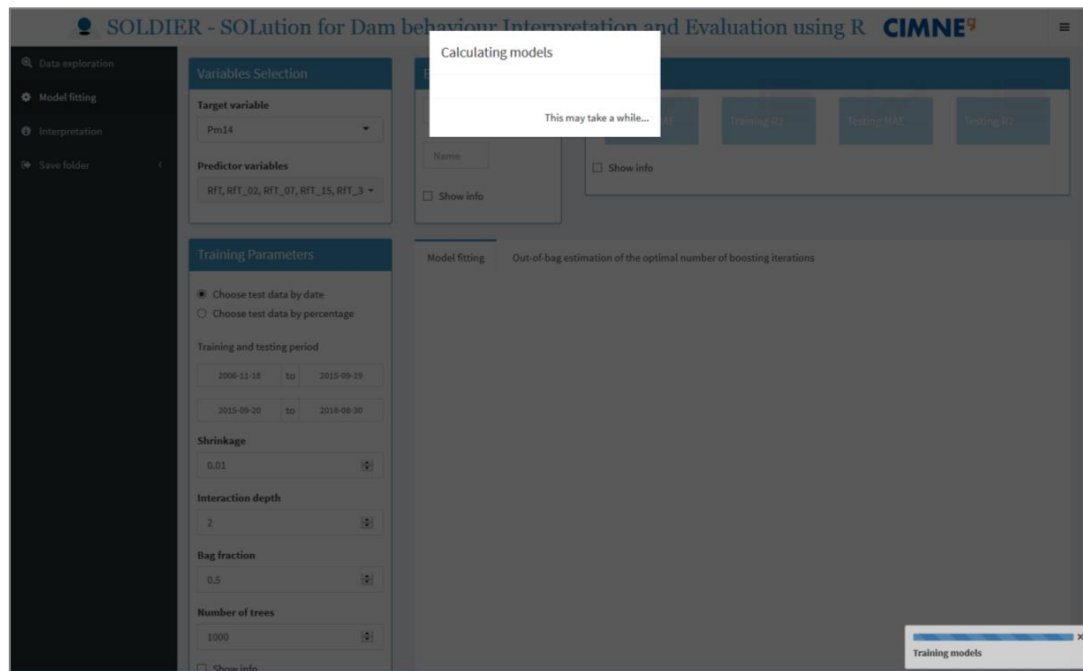

Figure 17.   Snapshot of the app while training models.

5.4   Model accuracy

The main goal of this app is interpreting dam response and the association between variables. This is done by analysing BRT predictive models for different outcomes. The discrepancy between model predictions and observations is computed and displayed, both for the training and test set, as the mean absolute error (MAE) and the coefficient of determination (R^2).
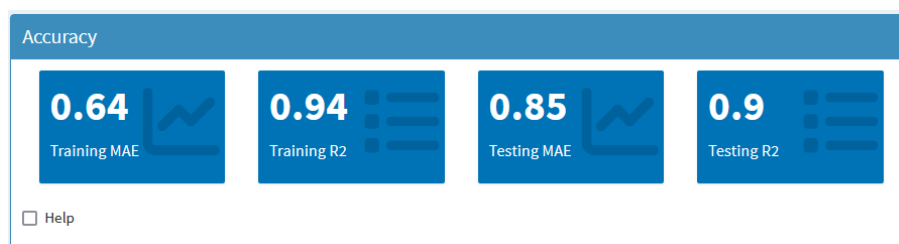

Figure 18.   Box to show model accuracy.

5.5   Predictions vs observations

The main box in this tab provides an overview of the regression fit for both calibration and validation periods (separated by a grey vertical line). It shows two time-series plots: the target variable together with the model predictions at the top, and the residuals (observation minus prediction) below. Both are interactive as the time series in the exploration tab.
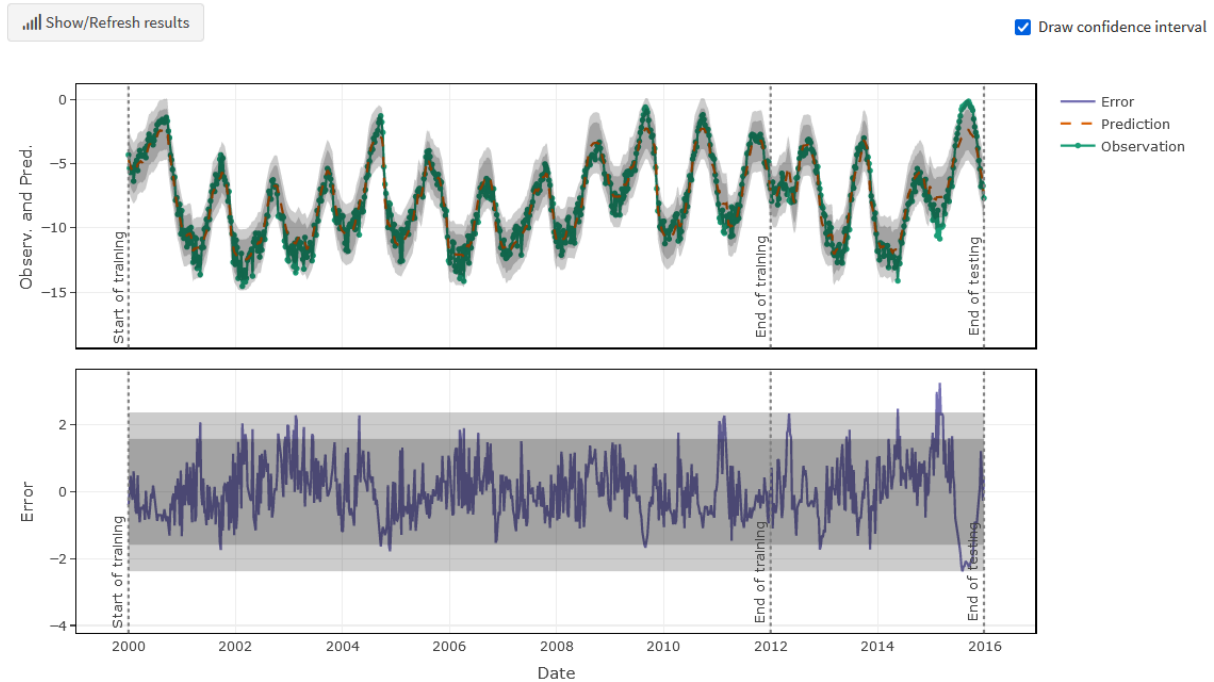
Figure 19.   Predictions vs observations.

The second subtab shows the optimal number of boosting iterations based on the out-of-bag data. In short, the vertical blue dotted line indicates the recommended number of iterations, i.e. "Number of trees" to be set for training. This plot is helpful to check if the training parameters used are appropriate. The curve typically has a sharp decrease followed by a horizontal asymptote. If the blue line coincides with the chosen value for the "Number of trees" (i.e. is located at the right of the plot), and the curve is far from the horizontal in that area, the model should be trained with more trees.

The effect of the number of trees is associated to that of the shrinkage coefficient: low values of shrinkage require more trees to obtain accurate predictions. Better accuracy for training can be obtained with high shrinkage and less trees, at the cost of increasing the potential for overfitting. Some authors [5] suggest setting a small shrinkage (0.001 or lower) and adding as many trees as necessary to reach the quasi-horizontal area in the error curve. Computational cost can be an issue in some settings, but that is not the case with the usual volume of data to be handled in this application.
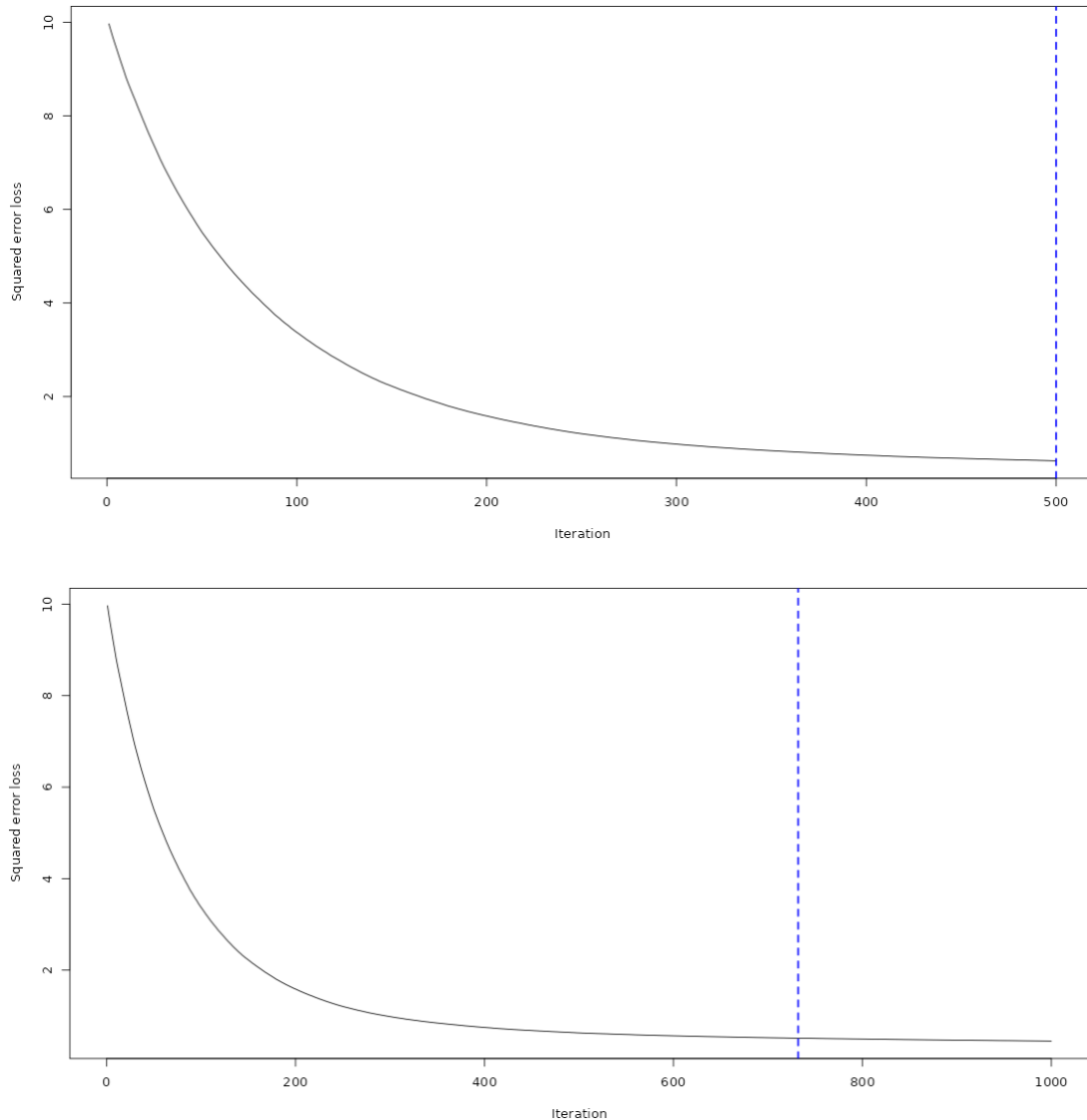
Figure 20.   Estimation of the optimal number of trees. In the top chart, the blue line is at the right end, and the error curve is not flat. This means that more trees are needed. In the example at the bottom, the curve is basically flat at the right of the dotted line; therefore, the amount of trees is appropriate. More precisely, it might be reduced in this case.

6.   TAB 3: Interpretation

6.1   Relative influence of predictors

In this box, the predictive model can be analysed through the relative influence plots. The first tab shows a bar graph with the 10 predictors with higher association with the target variable (in alphabetical order). The second tab shows the same results, but grouped by the type of input. This is useful, for instance, to compare the relative effect of temperature to that of the reservoir level, in case several variables depending on each load have been considered as inputs. In the example, the moving averages of air temperature over different periods were considered, so their importance is summed for the second plot.
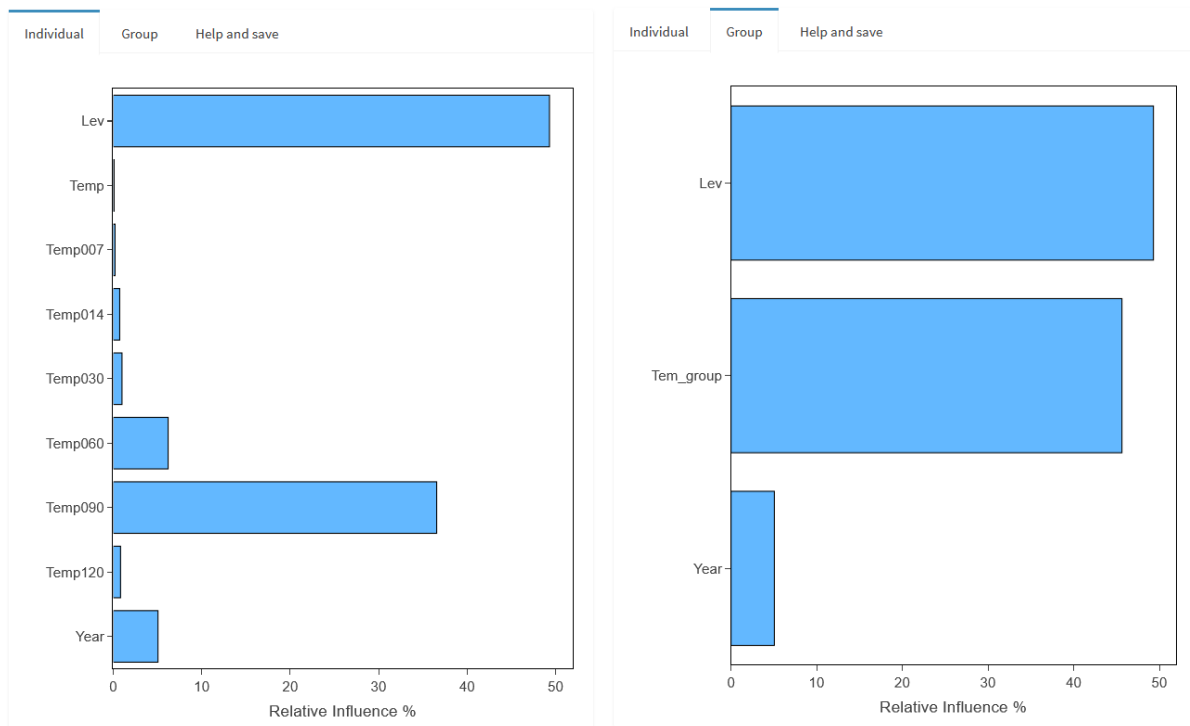
Figure 21. Relative influence. Left: value for the top variables. Right: same results by groups.

The third subtab offers the option to save the raw data of relative influence as a csv file.

Note: The plots can be saved or copied by some screen-shot utility.

6.2 Partial dependence

A partial dependence plot (PDP) shows the average prediction of the model when replacing the actual value of a given input by a set of equally-spaced values. This can be done for two of the inputs, selected from the drop-down menu, in the first subtab (Figure 22).
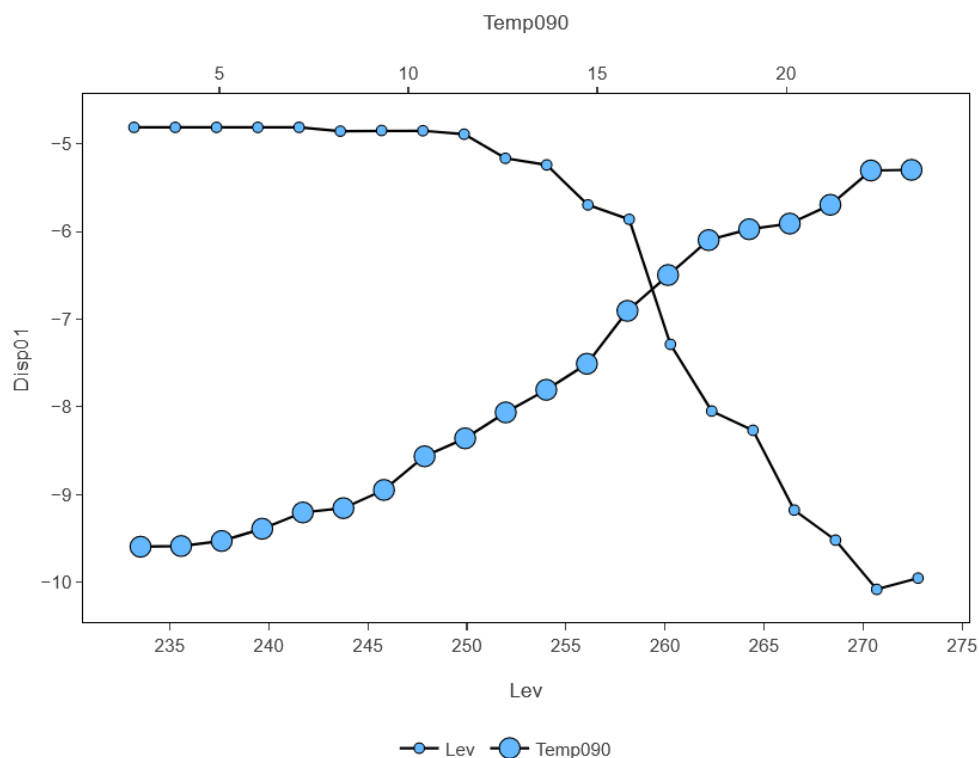
Figure 22. Example of partial dependence plot.

The same can be done for couples of inputs. In SOLDIER, such results can be explored either as a heatmap (Figure 23) or as a 3D plot (Figure 24).
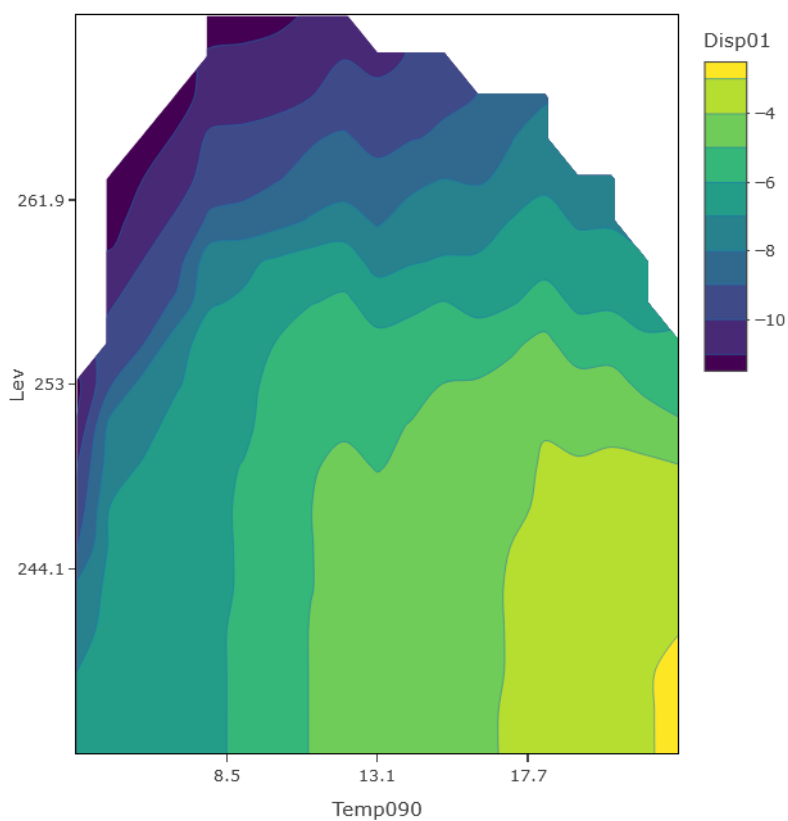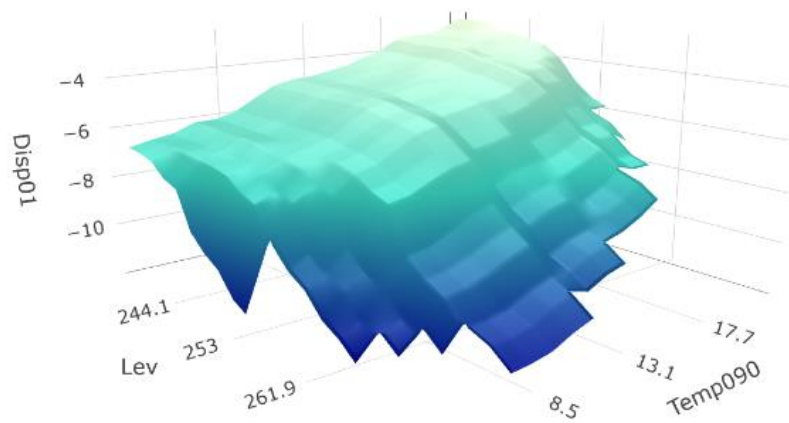

Figure 23. Heatmap of the combined influence of two inputs.

Figure 24.   3D surface of the combined influence of two inputs.

7. References

[1] R Development Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

[2] Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide to boosted regression trees. Journal of Animal Ecology, 77(4), 802-813.

[3] Salazar, F., Toledo, M. A., Oñate, E., & Morán, R. (2015a). An empirical comparison of machine learning techniques for dam behaviour modelling. Structural Safety, 56, 9-17.

[4] Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie and Jonathan McPherson (2017). shiny: Web Application Framework for R. R package version 1.0.5. https://CRAN.R-project.org/package=shiny

[5] Ridgeway, G. (2007). Generalized Boosted Models: A guide to the gbm package. Update, 1(1), 2007.