



COORDINATED PRODUCTION
FOR BETTER RESOURCE EFFICIENCY

D1.2 Prototype tool or prototype software components for construction of hybrid models

Maarten Nauta
PSE – United Kingdom

August 2019
www.spire2030.eu/copro



Project Details

PROJECT TITLE	Improved energy and resource efficiency by better coordination of production in the process industries
PROJECT ACRONYM	CoPRO
GRANT AGREEMENT NO	723575
INSTRUMENT	RESEARCH AND INNOVATION ACTION
CALL	H2020-SPIRE-02-2016
STARTING DATE OF PROJECT	NOVEMBER, 1ST 2016
PROJECT DURATION	42 MONTHS
PROJECT COORDINATOR (ORGANIZATION)	PROF. SEBASTIAN ENGELL (TUDO)

THE COPRO PROJECT

The goal of CoPro is to develop and to demonstrate methods and tools for process monitoring and optimal dynamic planning, scheduling and control of plants, industrial sites and clusters under dynamic market conditions. CoPro pays special attention to the role of operators and managers in plant-wide control solutions and to the deployment of advanced solutions in industrial sites with a heterogeneous IT environment. As the effort required for the development and maintenance of accurate plant models is the bottleneck for the development and long-term operation of advanced control and scheduling solutions, CoPro will develop methods for efficient modelling and for model quality monitoring and model adaption.

The CoPro Consortium

Participant No	Participant organisation name	Country	Organisation
1 (Coordinator)	Technische Universität Dortmund (TUDO)	DE	HES
2	INEOS Manufacturing Deutschland GmbH (INEOS)	DE	IND
3	Covestro Deutschland AG (COV)	DE	IND
4	Procter & Gamble Services Company NV (P&G)	BE	IND
5	Lenzing Aktiengesellschaft (LENZING)	AU	IND
6	Frinsa del Noroeste S.A. (Frinsa)	ES	IND
7	Universidad de Valladolid (UVA)	ES	HES
8	École Polytechnique Fédérale de Lausanne (EPFL)	CH	HES
9	Ethniko Kentro Erevnas Kai Technologikis Anaptyxis (CERTH)	GR	RES
10	IIM-CSIC (CSIC)	ES	RES
11	LeiKon GmbH (LEIKON)	DE	SME

Deliverable 1.2 Prototype tool or prototype software components for construction of hybrid models

12	Process Systems Enterprise LTD (PSE)	UK	SME
13	Divis Intelligent Solutions GmbH (divis)	DE	SME
14	Argent & Waugh Ltd. (Sabisu)	UK	SME
15	ASM Soft S.L (ASM)	ES	SME
16	ORSOFT GmbH (ORS)	DE	SME
17	Inno TSD (inno)	FR	SME

Document details

DELIVERABLE TYPE	REPORT	
DELIVERABLE NO	1.2	
DELIVERABLE TITLE	REPORT ON Prototype tool or prototype software components for construction of hybrid models	
NAME OF LEAD PARTNER FOR THIS DELIVERABLE	PSE	
VERSION	1.0a	
CONTRACTUAL DELIVERY DATE	31 JULY 2019	
ACTUAL DELIVERY DATE	12 SEPTEMBER 2019	
Dissemination level		
PU	Public	X
CO	Confidential, only for members of the consortium (including the Commission)	

Abstract

As part of the COPRO project, PSE has developed a prototype hybrid-modelling tool. This prototype tool facilitates the construction and evaluation of data-driven and hybrid models using the gPROMS modelling platform. It consists of a number of software components. This document describes these components, their implementation and the user workflow.

REVISION HISTORY

The following table describes the main changes done in the document since it was created.

Revision	Date	Description	Author (Organisation)
V1.0a	28/01/2020	Public deliverable modified from original deliverable	K.M. Nauta (PSE)

Disclaimer

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Any liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No license, express or implied, by estoppels or otherwise, to any

intellectual property rights are granted herein. The members of the project CoPro do not accept any liability for actions or omissions of CoPro members or third parties and disclaims any obligation to enforce the use of this document. This document is subject to change without notice.

Table of contents

1	Executive summary	7
2	Introduction	8
2.1	Introduction	8
3	Workflows	9
3.1	Overview	9
3.2	Workflows	9
3.2.1	Surrogate modelling	9
3.2.1	Including a data-driven model (part) in a gPROMS Simulation.....	10
3.2.1	Estimation/fitting of a hybrid model.....	11
4	ALAMO-type model generation tool	13
4.1	Introduction	13
4.2	Running the tool	13
4.3	Example execution	15
4.4	GUI design for surrogate modelling	15
5	Data-driven model component	19
5.1	Introduction	19
5.2	Example	20
6	Hybrid model library	22
6.1	Introduction	22
6.2	Unit operations	22
6.3	Example: Furnace in olefins plant	23
7	Bibliography	25

1 Executive summary

In the framework of the COPRO project PSE has developed a prototype hybrid-modelling tool that integrates with PSE's gPROMS process simulation software (Barton & Pantelides, 1993). Hybrid-modelling is a modelling approach whereby models are generated both from first-principle models (models that are based on prior knowledge about a process) and data-driven models (models derived directly from data). By combining these approaches the intention is to allow users to choose the most appropriate approach for each part of the model, thus permitting them to generate models in the most efficient manner. Hybrid modelling and its use in the COPRO project is described in more detail in the COPRO report for the D1.1 deliverable (COPRO, 2019).

The prototype hybrid-modelling tool allows the user to derive data-based and hybrid models from data and use these models directly in the existing gPROMS process simulation package. Together these components (partially) facilitate different three different types of workflow:

1. Surrogate modelling. This is where the outputs of a complex first-principle model are used to generate a data-driven model that can be used to rapidly predict the behaviour of the unit operation represented by the complex first-principle model.
2. Generation of a data-driven part of a model and including this part in a first-principles model or flowsheet to obtain a hybrid model.
3. Use process data to perform data-reconciliation using a first-principle model in order to obtain the inputs and outputs for a part of the process that will be modelled using a data-driven model. Then fit the data-driven part based on these inputs and outputs and integrate this part with the first-principle model to predict the outputs of the complete process.

This report describes the workflow in more detail and reports in detail on the individual software components that were developed.

2 Introduction

2.1 Introduction

PSE's existing gPROMS software environment (Barton & Pantelides, 1993) is well suited for first-principle modelling. It includes a modelling language to write such models, a robust&scalable kernel to evaluate such models, and a Graphical User Interface (GUI) referred to as the gPROMS Integrated Modelling Environment (IME) to let users develop the models and perform calculations easily. Currently the capabilities of this tool to perform data-driven modelling are limited. The aim of the hybrid modelling tool is to extend these capabilities and thereby enable hybrid and data-driven modelling within the gPROMS environment. The "prototype" hybrid modelling tool developed within the framework of the COPRO project currently consists of a number of software components, which are partially integrated in the gPROMS software product.

To generate hybrid and data-driven models with the prototype hybrid modelling tool, the user has a few alternative use-cases and corresponding workflows. The workflow paths are described in Section 3.

The software components are used as part of different workflow paths. They are:

1. ALAMO-type (Cozad, Sahinidis, & Miller, 2014) model generation tool (see Section 4)
2. Data-driven model component (see Section 5)
3. Hybrid model library (see Section 6)

This report describes the individual software components, their use and gives an example of each. When used as part of a workflow these software components need to exchange data with the user (input & output) and with other components. There are two dataformats that have been developed in this project for this purpose:

1. ALAMO model generation tool config file (json)
2. DataDrivenModel XML format.

These are described in more detail in 7 Appendix: Data Formats.

In the framework of the COPRO project also prototype Python code has been developed for the purpose of fitting data-driven models and storing these in the DataDrivenModel XML format. This code is not part of this deliverable as it is not likely to be productized.

3 Workflows

3.1 Overview

The prototype hybrid-modelling tool allows the user to derive data-based and hybrid models from data and use these models directly in the existing gPROMS process simulation package. Together these components (partially) facilitate different three different types of workflow:

1. Surrogate modelling. This is where the outputs of a complex first-principle model are used to generate a data-driven model that can be used to rapidly predict the behaviour of the unit operation represented by the complex first-principle model.
2. Generation of a data-driven part of a model and including this part in a first-principles model or flowsheet to obtain a hybrid model.
3. Use process data to perform data-reconciliation using a first-principle model in order to obtain the inputs and outputs for a part of the process that will be modelled using a data-driven model. Then fit the data-driven part based on these inputs and outputs and integrate this part with the first-principle model to predict the outputs of the complete process.

3.2 Workflows

3.2.1 Surrogate modelling

Surrogate modelling is an approach whereby behavior relevant for a particular application of an existing complex first-principle model is captured in an efficient data-driven model, which is referred to as the “surrogate” model. That surrogate model is then use for the particular application. The advantage of using this approach as compared to using the first-principles model directly is that evaluations of the surrogate model are very rapid. This means the surrogate model can be used for purposes where use of the first-principles model might be prohibitive. Examples of this are including the model as part of other (very complex) flowsheets where many instances of it may be present, using it as part of complex calculations (non-linear optimization) where many evaluation iterations are required or using it for online applications where real-time computation is required.

The workflow (see Figure 1) for accomplishing this will be implemented fully in the gPROMS IME and all steps in the workflow can be performed using a Graphical User Interface. It involves 3 steps:

1. Global System Analysis (existing functionality in gPROMS Platform + IME) to generate input output maps from a given first-principles model.
2. Surrogate model fitting using the ALAMO approach (algorithm implemented as part of the developments in the COPRO project)
3. Evaluation in gPROMS (when required in conjunction with other first-principle models).

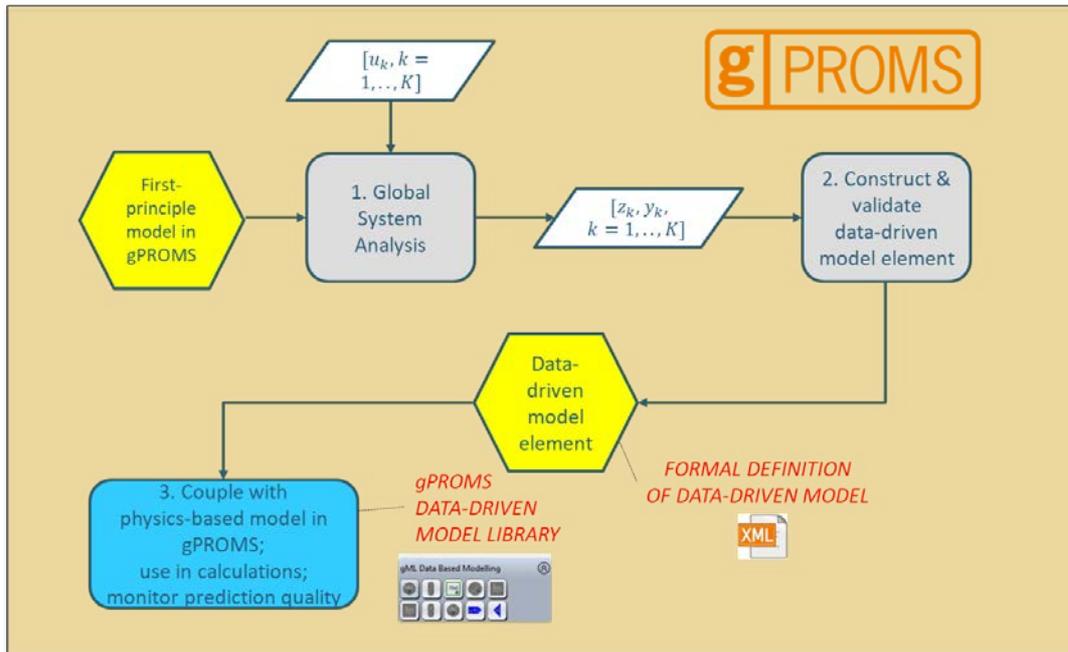


Figure 1 Workflow for surrogate modelling

3.2.1 Including a data-driven model (part) in a gPROMS Simulation

In some cases deriving a first-principle model for a particular part / feature of a process or model can be difficult. Examples of this are poorly understood phenomena (e.g. fouling) or very complex unit operations / plants. In this case, a data-driven model can be derived directly from plant/experiment data for this part of the process and can then be combined with first-principle models for the rest of the process. For this workflow we consider the situation where the inputs and outputs for the data-driven part are directly “measurable” (i.e. they are in the dataset). The workflow (see Figure 2) for this can be accomplished partially in the gPROMS IME and contains the following steps:

1. Data pre-processing / data-analysis
2. Construct and validate a data-driven model
3. Evaluation in gPROMS (when required in conjunction with other first-principle models).

The first 2 steps are not part of the prototype tool but can be conducted with any data-driven modelling tool (e.g. Python+sk-learn (Pedregosa, 2011)).

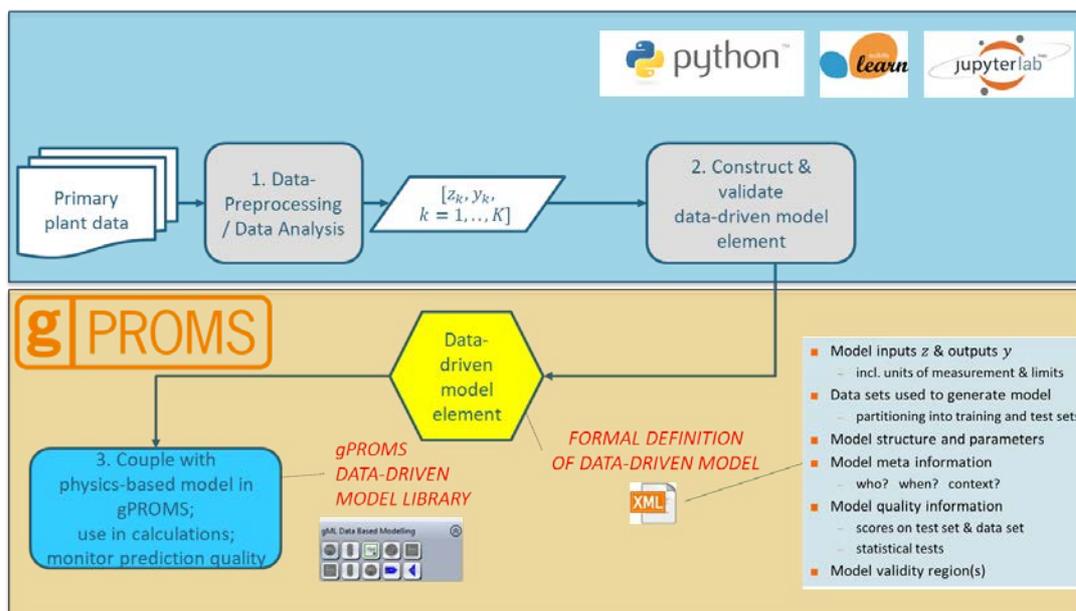


Figure 2 Workflow for including a data-driven component in a gPROMS simulation

3.2.1 Estimation/fitting of a hybrid model

This use-case is similar to the previous one but for one difference: not all of the inputs and outputs for the data-driven part are directly “measurable” (i.e. they are in the dataset). For this case the workflow contains an additional step: data-reconciliation. In this step a first principle model is used to generate estimates for the inputs and outputs of the data-driven part of the model based on quantities that can be measured. The workflow (see Figure 3) for this can be accomplished partially in the gPROMS IME and contains the following steps:

1. Data pre-processing / data-analysis
2. Data-reconciliation using a first-principle model
3. Construct and validate a data-driven model
4. Evaluation in gPROMS (when required in conjunction with other first-principle models).

The first 3 steps are not part of the prototype tool but can be conducted with any data-driven modelling tool (e.g. Python+sk-learn (Pedregosa, 2011)). For the data-reconciliation step the existing gPROMS gO:RUN functionality is used.

Deliverable 1.2 Prototype tool or prototype software components for construction of hybrid models

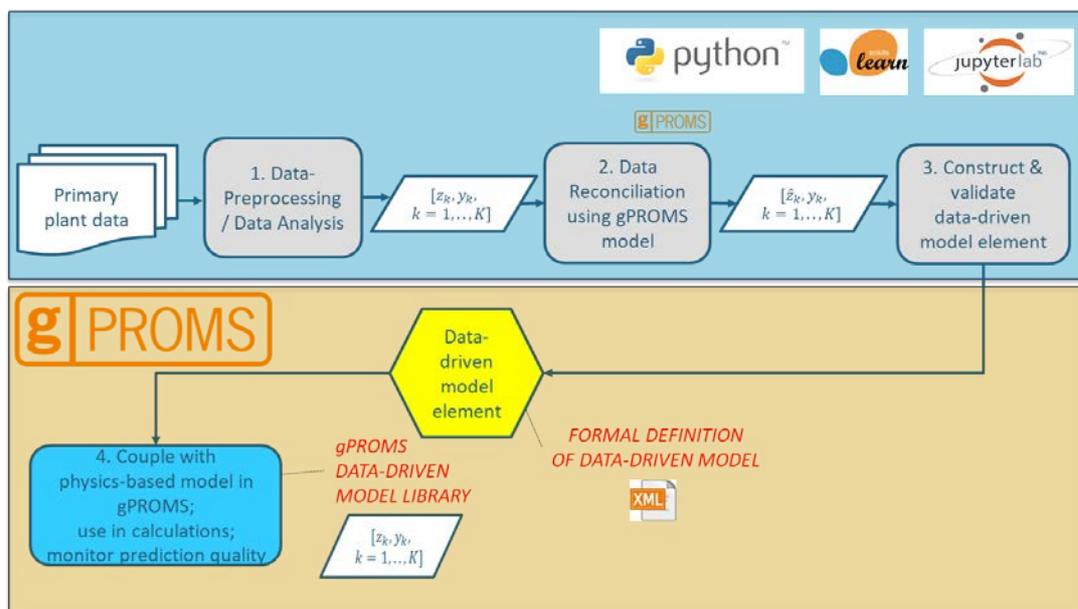


Figure 3 Workflow for estimation/fitting of a hybrid model

4 ALAMO-type model generation tool

4.1 Introduction

This tool can be used to generate data-driven models using the ALAMO approach (Cozad, Sahinidis, & Miller, 2014). It is a standalone application (SurrogateModeller.exe) that runs in a command shell. It generates data-driven models based on a configuration file (in JSON format, see Appendix 7.1) that the user supplies. The configuration file specifies among other things what data source to use, which variables to treat as inputs and outputs and information related to the model generators. Currently, only one model generator is supported: the ALAMO model generator.

When the application runs, it reads the configuration file, loads the data and invokes the specified model generator. For each specified output, the generator will calculate the appropriate data-driven models; progress is displayed as the models are calculated. The final form of the models can be displayed; an XML representation of the models is also written to disk.

Component	Surrogate
Type	Native C++ DLL + EXE
Language	C++14
Input format	json, gPROMS data-model
Output format	XML
Software Licensing	gPROMS licensing Xpress Solver suite licensing

4.2 Running the tool

Executing SurrogateModeller.exe without any argument or with the `-h` argument, prints the help screen:

```
SurrogateModeller.exe -h
```

```
-----  
Surrogate Modelling Tool  
-----  
  
Version: 1.0.0  
  
Allowed options:  
  
General options:  
  -h [ --help ]           Help screen  
  -v [ --version ]       Version  
  
Shell options:  
  -i [ --interactive ]   Interactive shell  
  
Generation options:  
  -c [ --config ] arg    Configuration file
```

Figure 4: SurrogateModeller Help screen

Passing the `-i` flag starts the tool in interactive mode; this allows the user to supply configuration files one at a time with the tool generating models each time until the user exits by pressing `Q`.

```
-----  
Surrogate Modelling Tool  
-----  
  
>>> Path to configuration file (q/Q/quit to exit)...
```

Figure 5: SurrogateModeller interactive mode

Whereas using the `-c` flag causes the tool to generate models for a single configuration and then exit.

Figure 6: Example configuration file

4.3 Example execution

The following figure shows an example of the output generated by the tool.

```
-----  
Surrogate Modelling Tool  
-----  
  
Loading configuration file: hump6.json...  
Loaded "pse-sm-alamod.dll".  
  
-----  
Configuration Summary  
-----  
Data source:          Hump6.csv  
Inputs:  
    x0:  x1  
    x1:  x2  
Outputs:  
    z0:  z  
Input Scaling:      NONE  
Output Scaling:     NONE  
Number of Points:  
    Data: 121  
    Training: 121  
# Basis Functions:  21  
Max # of terms:    21  
  
Model for output: z  
Level  Objective Value    Fitness Metric  Root-Mean-Sq Err  
  0      1673.0156         734.23558     1.8733583  
  1      287.61706         278.24062     0.28221395  
  2      213.24187         188.63209     0.19319314  
  3      108.23096          37.60053     0.10259112  
  4       69.450714        -45.965797    0.071983753  
  5       44.6184          -174.35399    0.041961704  
  6      -6.0915831e-11     -5455.664     1.3833771e-11  
  
Final model details:  
  Level selected: 6  
  Reason: fit is good enough  
  
  Solution: z = 4*x0^2 - 4*x1^2 - 2.1*x0^4 + 4*x1^4 +  
              0.333333*x0^6 + 1*x0.x1
```

Figure 7: SurrogateModeller example output

First a summary is printed of the main input parameters. Then, for each output, the solution metrics (objective function, the fitness metric and the root mean square error) at each iteration level are displayed.

Lastly, the actual solution is displayed, in terms of the scaled inputs – and the scaling of those inputs is also displayed.

This information is also written to a text file so a permanent record is kept. An XML file is also produced for further consumption by internal PSE components.

4.4 GUI design for surrogate modelling

As part of the COPRO project a GUI for surrogate modelling has been designed which will become part of the gPROMS IME. It integrates with the Global System Analysis functionality in the gPROMS

IME. This GUI is currently in the process of being implemented. Screenshots from a beta version are shown in this report.

When a Global System Analysis (GSA) (existing gPROMS functionality) activity has been performed a “case file” is generated that can be accessed in the gPROMS IME. This case file will have a new category “Surrogate Models”, right-clicking on which will permit the user to generate a new surrogate model (see Figure 8). When this option is select a new Surrogate Model entity will be generated. On accessing this, a tabbed window is shown where the surrogate model generation can be configured. On the first tab “General” (see Figure 9) the inputs and outputs of the surrogate model can be selected from the available inputs and outputs contained in the data generated by the GSA activity. On the “Algorithm” tab (see Figure 9) the type of machine learning algorithm can be selected (currently only “ALAMO” is supported) and then the parameters for this algorithm (for ALAMO these are the basis functions and the numerical solution parameters) can be configured.

The user can then execute the activity and obtain an HTML report. The predictions for the generated surrogate model will then be integrated with the rest of the data for the GSA.

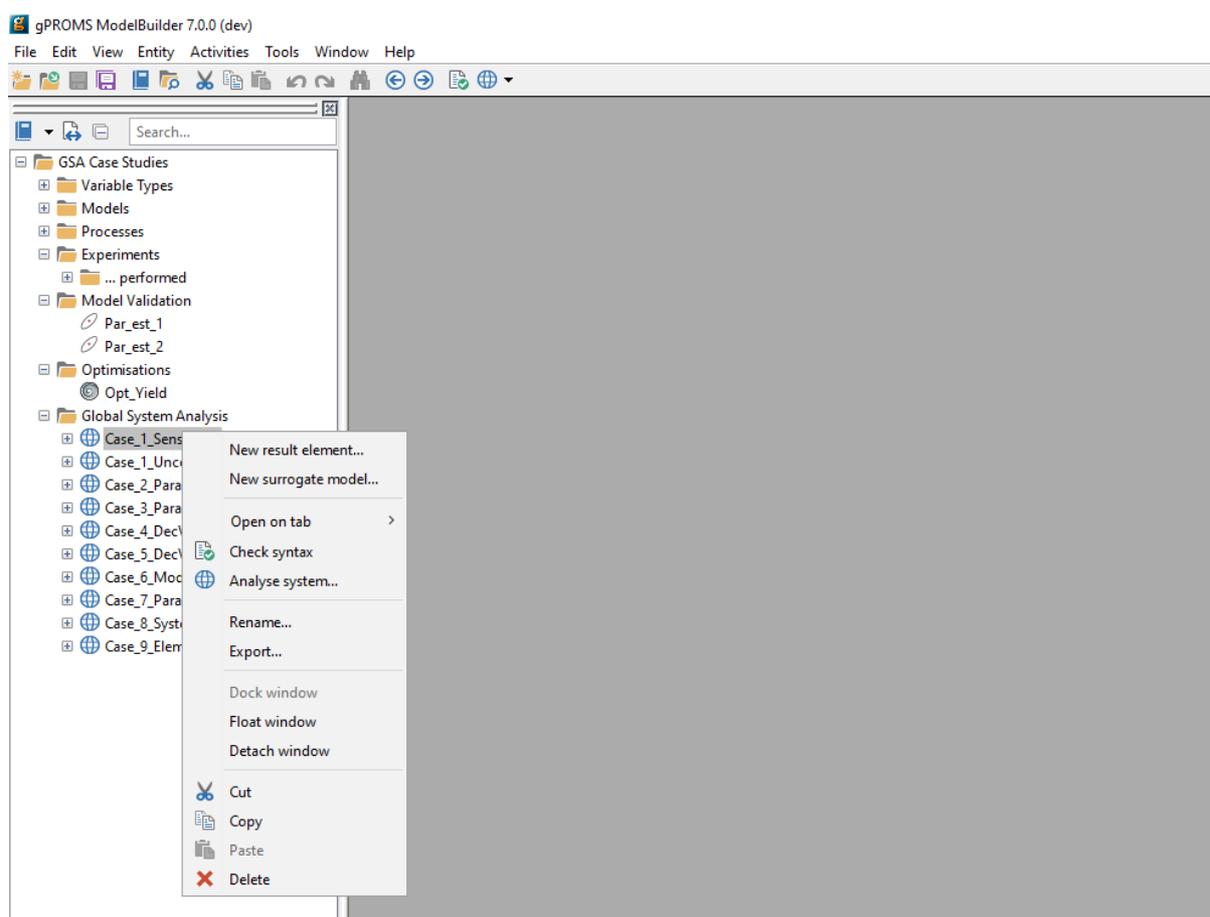


Figure 8: “New surrogate model” option in GSA context menu (GSA Case project)

Deliverable 1.2 Prototype tool or prototype software components for construction of hybrid models

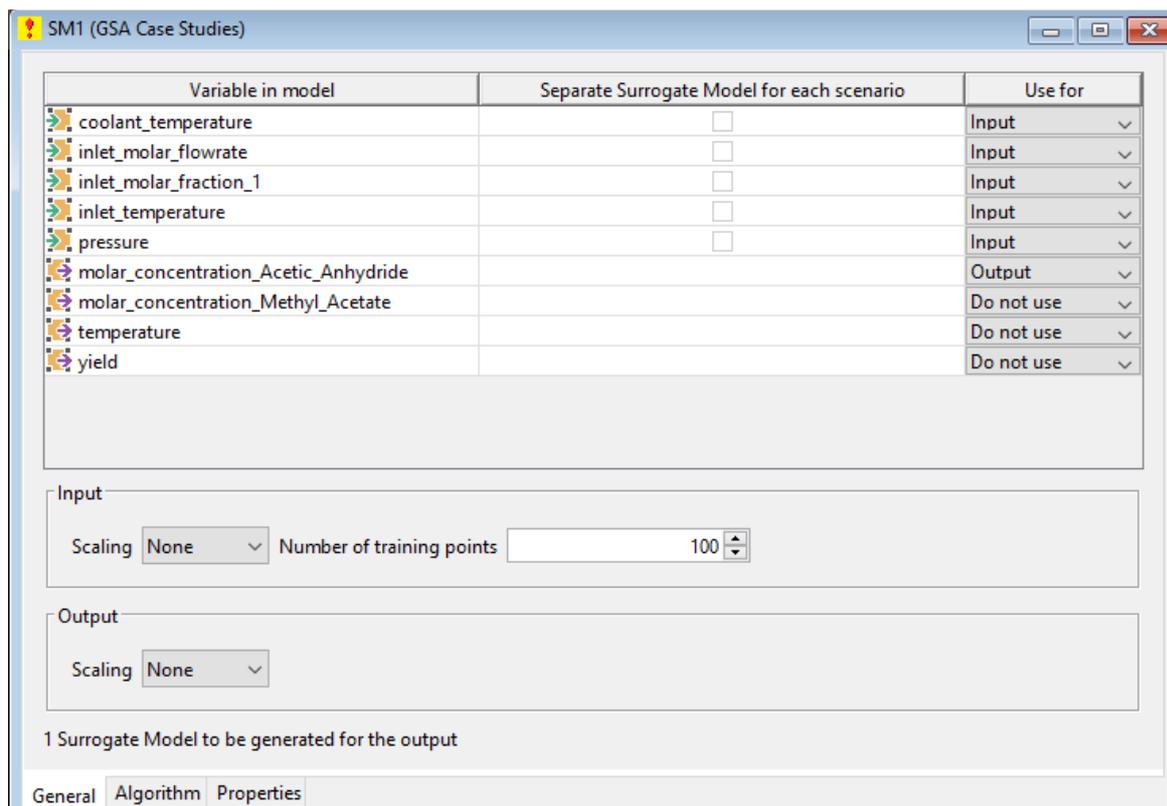


Figure 9 "General" tab for surrogate model configuration

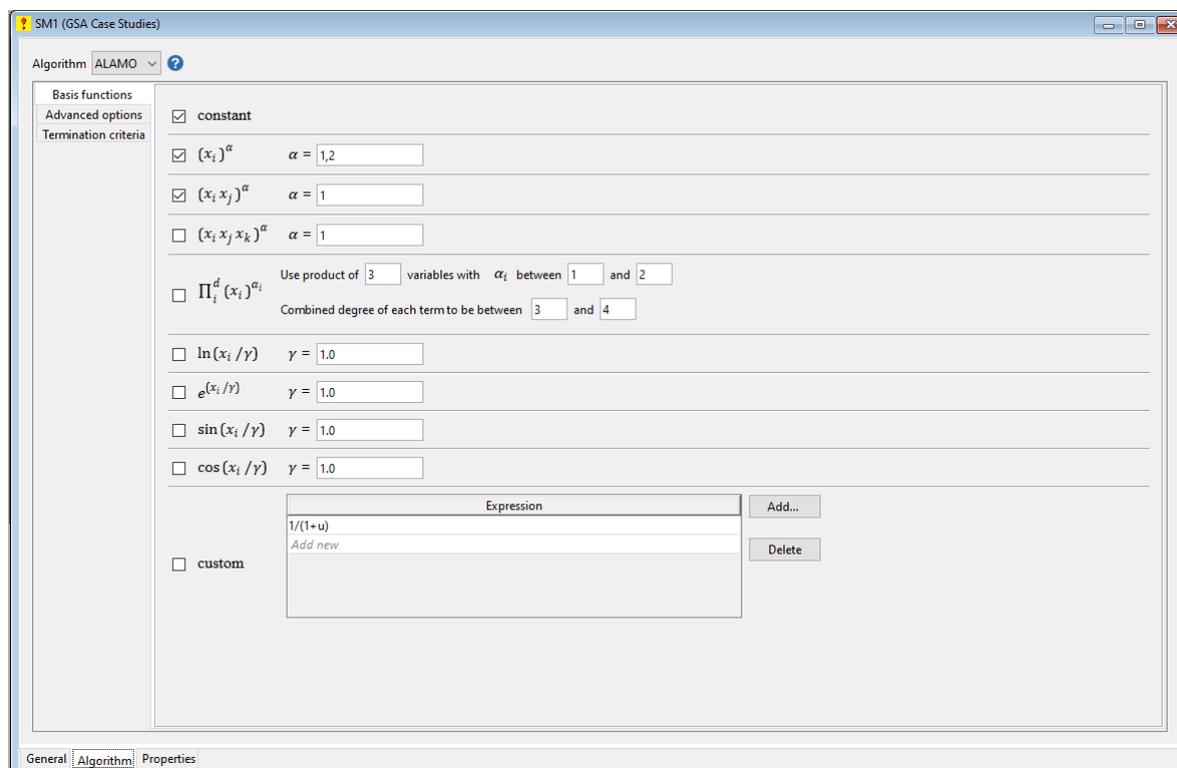


Figure 10 Algorithm tab of the surrogate modelling GUI

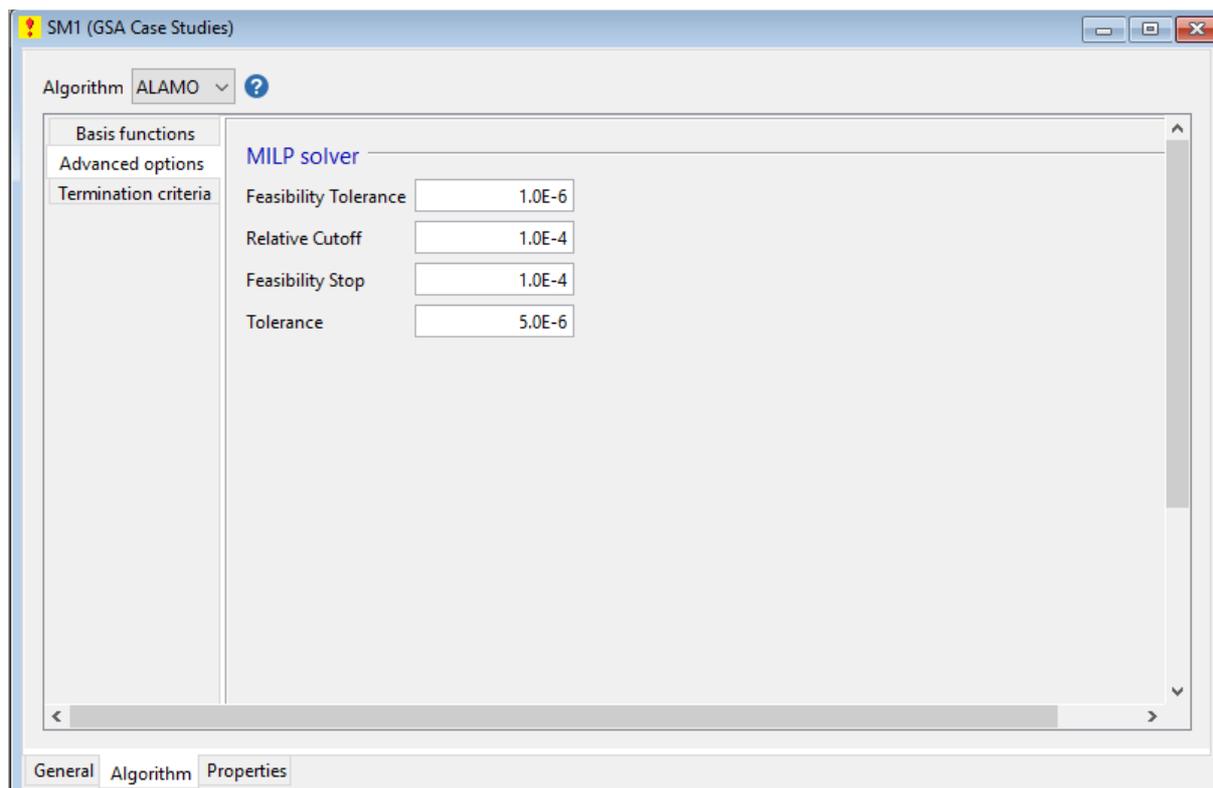


Figure 11 Solution parameters for the algorithm tab of the surrogate modeling GUI

```

Execution Output (Case_1_Uncertainty_20200128_101009)
Show messages to level 2
Requesting license for 0245_GSA_SURROGATE from server ...
License granted by server licsserv1.psenterprise.com.
Surrogate Model Generation
  Generating models for: SM1
    Loaded "pse-sm-alamo.dll".
    Requesting license for 0251_XPRESS from server ...
    License granted by server <local license file>.

-----
Configuration Summary
-----
Data source:          Case_1_Uncertainty.nc
Inputs:
    u0:  coolant_temperature
    u1:  inlet_molar_flowrate
    u2:  inlet_molar_fraction_1
    u3:  inlet_temperature
    u4:  pressure
Outputs:
    y0:  molar_concentration_Acetic_Anhydride
Input Scaling:      NONE
Output Scaling:     NONE
Number of Points:
    Data: 100
    Training: 100
# Basis Functions:  21
Max # of terms:    21

Model for output: molar_concentration_Acetic_Anhydride [case 0]
Level  Objective Value  Fitness Metric  Root-Mean-Sq Err
0      1.1174754         325.16651     0.50310488
1      1e+40              381.50179     0.67362874
2      1e+40              381.50179     0.67362874
3      1e+40              381.50179     0.67362874
4      1e+40              381.50179     0.67362874
5      1e+40              381.50179     0.67362874
    
```

Figure 12 Execution output from estimating a surrogate model from GSA data

5 Data-driven model component

5.1 Introduction

The role of the data-driven model component is to integrate the predictions of data-driven models in the gPROMS Process Modelling package. This is accomplished by reading in a DataBasedModel XML file and then using the gPROMS Foreign Object Interface to receive inputs and generate predictions for that model.

Component	DataBasedModelFO
Type	Native C++ DLL
Language	C++17
Input format	DataBasedModel XML

Output format	gPROMS Foreign Object interface
Software Licensing	gPROMS Licensing

5.2 Example

Figure 12 shows the use of the predict, constraints and warn methods in the gPROMS modelling language. Figure 13 shows the execution of the predict, constraints and warn methods by the gPROMS kernel.

```

DataBasedModel_names_gML (gML Data Based Modelling Foreign Object)
113 current_input_information_types := current_value_OS + input_
114 current_output_information_types := current_value_OS + output_
115
116 EQUATION
117 # Prediction of outputs
118 y      = my_DataBasedModelFO.predict(x);
119
120 # Warnings for extrapolation
121 FOR i:=1 TO use_warnings DO
122     warning_status(i) = my_DataBasedModelFO.warn(x);
123 END
124
125 # Constraints for the purpose of optimisation
126 FOR i:=1 TO use_constraints DO
127     c() = my_DataBasedModelFO.constraints(x);
128 END
129
130 # Statistical tests
131 t2      = 0; my_DataBasedModelFO.t2(x);
    
```

112:1 INS

Interface Interface language Topology gPROMS language Properties

Figure 13 Example of the use of the driven modelling FO methods in the gPROMS modelling language

```
□ Performing precalculation
  IF condition ((Flowsheet.cs_custom.MB.F_out(2){0} - 1e-09){-1e-09} > 0){0} no longer satisfied.
  Object : DataBasedModelFO::acetone_methanol_water.xml
  Method : PREDICT
  Inputs :
  Input  5.0000000000000000E-01  7.4904458598726109E-01  5.0000000000000000E-01
  Outputs :
  7.3728477393530978E-01  -4.2265128838657647E+03  0.0000000000000000E+00  9.0251393271546937E-03
  Object : DataBasedModelFO::acetone_methanol_water.xml
  Method : WARN
  Inputs :
  Input  5.0000000000000000E-01  7.4904458598726109E-01  5.0000000000000000E-01
  Outputs :
  0.0000000000000000E+00
  Object : DataBasedModelFO::acetone_methanol_water.xml
  Method : CONSTRAINTS
  Inputs :
  Input  5.0000000000000000E-01  7.4904458598726109E-01  5.0000000000000000E-01
  Outputs :
  8.000000000000000016E-02  4.9044585987261136E-02  -1.0000000000000000E+00  1.9999999999999996E-01  2.
  Precalculation completed.
□ Performing initialisation calculation at time: 0
  Initialisation calculation completed.
  Execution of test_calc_spec_data_based completed successfully.

  Simulation took 1 seconds.
  Total CPU time: 1.234s (13% system time)

□ Performing Foreign Object termination: "DataBasedModelFO::acetone_methanol_water.xml"
  Foreign Object termination completed successfully.
```

Figure 14 Example of the use of the datadriven modelling FO in the gPROMS kernel

6 Hybrid model library

6.1 Introduction

The hybrid modelling library is a flowsheeting library of common process unit operations that can be included in a gPROMS ProcessBuilder flowsheet. The library supports drag&drop flowsheeting and configuration of unit operations using specification dialogs (as is customary in the gPROMS IME).

These models contain a first-principles part in the form of mass balances. The unit “performance” of each of these models is predicted by a data-driven part. The models used in the data-driven part can either be coded in the gPROMS modelling language as a series of mathematical expressions, or the Data-driven model component (see section 5) can be used to obtain predictions. The models are compatible with PSE’s gPROMS Model Libraries (gML).

Component	gML DataBasedModel
Type	gPROMS library project file(s) (.gPJ-PB)
Language	gROMS Modelling Language gPROMS Unit Meta Specification language
Input format	User specification via gPROMS IME
Output format	N/A
Software Licensing	gPROMS licensing

6.2 Unit operations

The units current in the library are listed in Table 1. The palette for dragging and dropping these units onto the flowsheet in the gPROMS IME is shown in Figure 14.

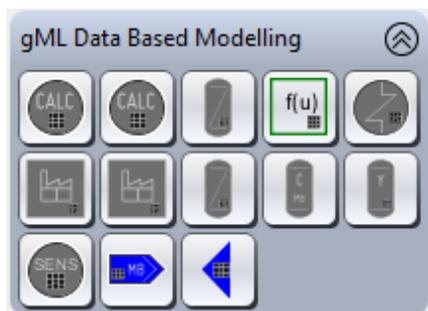


Figure 15 Flowsheet models in the hybrid model library

Table 1 Units in the hybrid model library

Model	Description
Calc	General calculation model that can be coupled to any gML unit operation model to allow variables in this model to be predicted from a data-driven

	model.
Component_splitter	Represents a general separation unit operation. A single inlet stream is split into N outlet streams and then split factors specified for each component in the inlet stream determine how much of that component ends up in each outlet stream.
Plant	Represents a general unit operation or high-level plant model. Mass balances are included in this model.
Plant_no_balance	Represents a general unit operation or high-level plant model. No mass or energy balances are enforced in this model.
Sensor	Represents a (soft)-sensor model. This model can be linked to any gML unit operation or stream and be used to predict quantities based on these inputs.
Source	Represents a material source on a flowsheet.
Reactor_conversion	Represents a conversion reactor.
Reactor_yield	Represents a yield reactor.
Heat exchanger	Represents a heat exchanger unit operation.

6.3 Example: Furnace in olefins plant

As part of the COPRO project a case study was conducted whereby a surrogate model was derived to represent a first-principle model of an Olefins furnace. This surrogate model was then included in a gPROMS flowsheet that represents a complete olefins plant using the hybrid model library (see Figure 15). Double-clicking the icon that represents the unit operation on the flowsheet brings up the specification dialog, see Figure 16, which permits configuring the unit operation and inspecting the model quality information and the model meta information for the data-driven part.

Deliverable 1.2 Prototype tool or prototype software components for construction of hybrid models

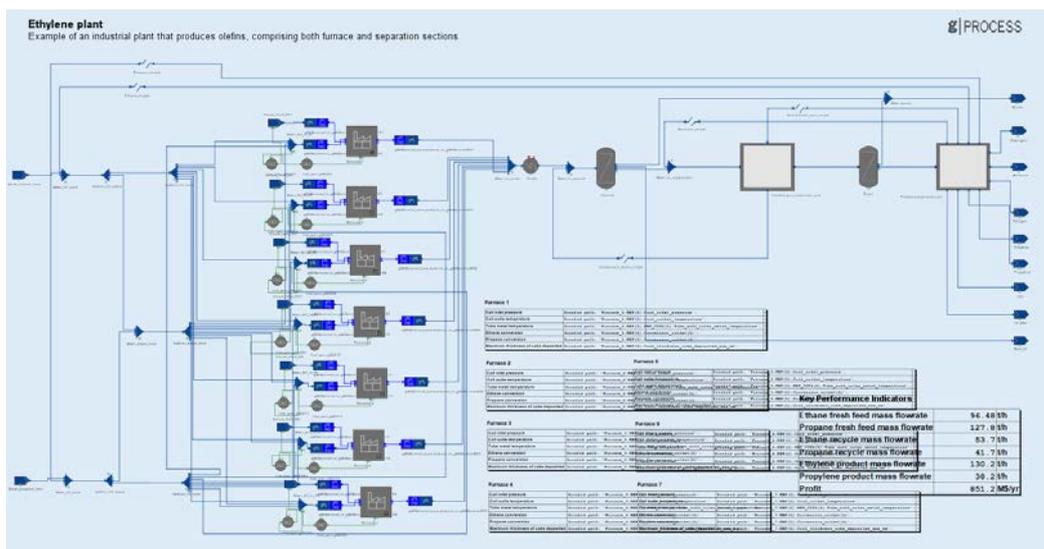


Figure 16 Flowsheet of an olefins plant with a data-driven unit operation model from the hybrid model library to represent the furnace

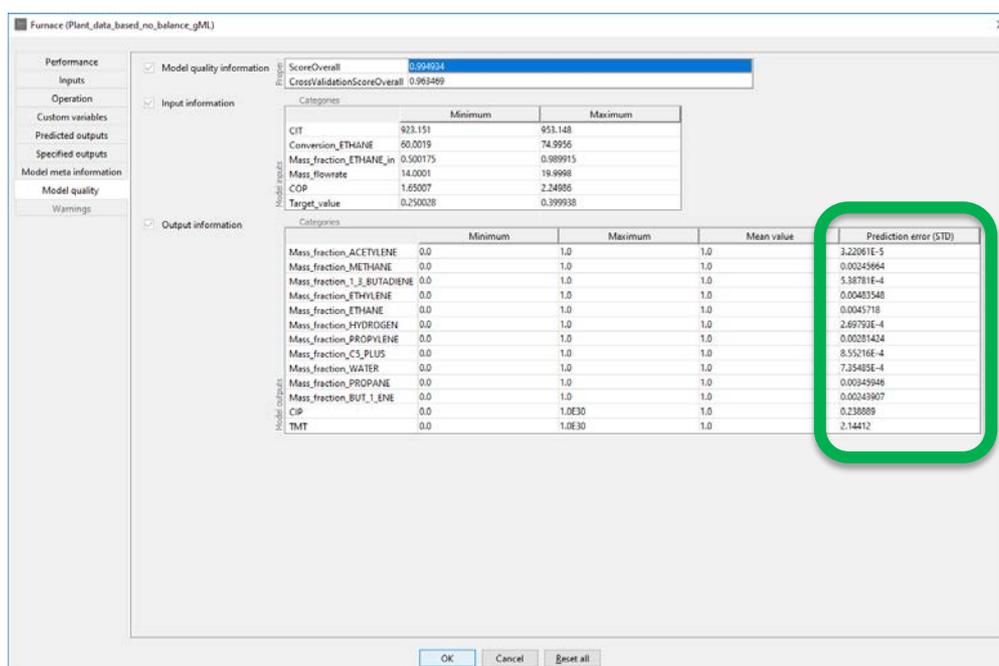


Figure 17 Presenting the user with output uncertainty estimations in PSE's hybrid modelling toolbox. For each output an estimate of the prediction error for unseen data is available based.

7 Bibliography

- Barton, P., & Pantelides, C. (1993). gPROMS - A combined discrete/continuous modelling environment for chemical processing systems. *Simulation Series*, 25, 25-34.
- COPRO. (2019). *D1.1 Efficient construction of efficient models*. COPRO.
- Cozad, A., Sahinidis, N. V., & Miller, D. (2014). Learning surrogate models for simulation-based optimization. *AIChE Journal*, 60(6), pp. 2211-2227.
- Pedregosa, F. e. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 2825-2830.