COORDINATED PRODUCTION
FOR BETTER RESOURCE EFFICIENCY

D1.3 Report on model quality monitoring, model uncertainty quantification, and model maintenance

Maarten Nauta (PSE), Christophe Fousette (Divis), Keivan Rahimi-Adli (INEOS), **Patrick Schiermoch (INEOS),** Benedikt Beisheim (INEOS), **Lukas Samuel Maxeiner (TUDO), Simon Wenzel (TUDO)**

December 2018
www.spire2030.eu/copro

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723575

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 Köln 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 Féderale 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

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	**1.3**	
DELIVERABLE TITLE	**REPORT ON MODEL QUALITY MONITORING, MODEL UNCERTAINTY QUANTIFICATION, AND MODEL MAINTENANCE**	
NAME OF LEAD PARTNER FOR THIS DELIVERABLE	**PSE**	
VERSION	**0.4**	
CONTRACTUAL DELIVERY DATE	**31 OCTOBER 2018**	
ACTUAL DELIVERY DATE	**19 DECEMBER 2018**	
Dissemination level		
PU	Public	X
CO	Confidential, only for members of the consortium (including the Commission)	

Abstract

As part of the COPRO project, models for industrial processes are developed and used for optimization of the processes. When models are developed and used it is important for users of the models to know whether the predictions from the models can be trusted. Therefore, this workpackage is concerned with model quality and model uncertainty.

This report presents a summary of the analysis that PSE and Divis conducted on how model quality is established during the model generation process, measures that are commonly used to quantify model quality and factors that influence these measures in practise. These measures generally signify how accurately a model will predict outputs of interest for previously unseen inputs. It also describes the ways model quality information is presented to the user of modelling tools, in this case PSE's hybrid modelling tool (developed as part of the COPRO project), INEOS Best Demonstrated Practise (BDP) toolbox (extended as part of the COPRO project) and Divis ClearVu analytics toolbox.

In this report quantifying model uncertainty is also considered. The concept of model uncertainty is closely related to model quality as the main aspect which determines the quality of a model is how accurately it will predict outputs for given inputs. The less accurate these predictions are the larger the uncertainty when the model is used.

Finally, this report considers model maintenance. Guidelines are given for maintaining models in an organization and aspect that influence maintenance are discussed (model serialization / file storage formats, model meta information, accessibility of models). This aspect of the report has a strong link to COPRO Deliverable D5.5: Requirement Specification and Functional Design Specification of the COPRO Model Management Platform.

REVISION HISTORY

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

Revision	Date	Description	Author (Organisation)
V0.1	05/10/2018	Initial draft by PSE with emailed contributions from Divis, TUDO, INEOS. Approval for public publication from Lenzing.	K.M. Nauta (PSE)
V0.2	16/10/2018	Revised draft after comments from reviewer TUDO. Conclusions and recommendations section added.	K.M. Nauta (PSE)
V0.3	22/10/2018	Revised after comments TUDO	K.M. Nauta (PSE)
V0.4	28/11/2018	Minor corrections TUDO	S. Engell, S. Wenzel (TUDO)
V0.4	19/12/2018	Final approval	S. Engell (TUDO)

Table of contents

1 Executive summary

As part of the COPRO project models for industrial processes are developed and used for optimization of the processes. When models are developed and used it is important for users of the models to know whether the predictions from the models can be trusted. Therefore, this workpackage is concerned with model quality and model uncertainty.

This report presents a summary of the analysis that PSE and Divis conducted on how model quality is established during the model generation process, measures that are commonly used to quantify model quality and factors that influence these measures in practise. These measures generally signify how accurately a model will predict outputs of interest for previously unseen inputs. It also describes the ways model quality information is presented to the user of modelling tools, in this case PSE's hybrid modelling tool (developed as part of the COPRO project), INEOS Best Demonstrated Practise (BDP) toolbox (extended as part of the COPRO project) and Divis ClearVu analytics toolbox.

The concept of model uncertainty is closely related to model quality. Model uncertainty together with model input uncertainty signifies how accurate a model will predict under certain conditions. Practical approaches to determine this uncertainty and work with it are discussed.

Finally, this report considers model maintenance. Guidelines are given for maintaining models in an organization and aspects that influence maintenance are discussed (model serialization / file storage formats, model meta information, accessibility of models). This aspect of the report has a strong link to COPRO Deliverable D5.5: Requirement Specification and Functional Design Specification of the COPRO Model Management Platform.

2 Model quality

2.1 Introduction

Model quality is first encountered during the model generation process. Measures for model quality are typically employed here to determine whether a particular model is "good enough" for use or to aid in the selection between different model architectures or values of hyper-parameters for a given model architecture. Therefore, in this Chapter this model generation process is reviewed to determine how concepts of model quality relate to it. The endpoint of the model generation process is typically a model with a desired quality according to a chosen measure.

As part of this document, data pre-processing is also briefly covered. The reason for this is that in this pre-processing step the modeller can choose to exclude/manipulate data that is deemed difficult to predict by the chosen modelling approach. When the model is then generated, model quality measures are applied to the model together with the pre-processed dataset (instead of the original dataset). Therefore, the pre-processing will affect the perceived model quality.

An important part of a discussion on model quality is how to convey the relevant information to the user of a modelling tool. As part of the COPRO-funded development of PSE's modelling tool a UI element has been designed that presents a summary of model quality information for any data-driven model included on a gPROMS (Process Systems Enterprise (PSE) Lmtd., 1997-2018) flowsheet.

Next, the monitoring of the model quality is discussed: how can the user be informed of the quality of predictions for "new" data? There are different scenarios for monitoring, depending on whether the model is used to predict batches of new data or single samples and whether measured outputs for this new data are available or not. Three industrial case studies from the COPRO project are used to illustrate how the model quality is considered in these case studies: a cooling tower data-driven model (Lenzing, Divis), a granulation soft-sensing example (PSE) and models for NH_3 network optimisation (INEOS, TUDO).

In addition to measurable concepts of model quality, there are also certain aspects of how a model is used in an organisation that will influence the level of trust users have in the model (perception of model quality). In particular, model meta information might influence this level of trust. For example, if it is known that a model was derived recently by a colleague who's expertise in relation to the process that is modelled, is trusted, a user might place more trust in the model. This aspect is not considered here but discussed further on in this document, namely Section 4.2.

Throughout this document it will be assumed that the model being generated is either a "first-principle" or "data-driven" model. Where there are differences between both modelling approaches in terms of workflow or model quality measures employed, this will be indicated.

This document only covers regression and not classification. It is also restricted on supervised learning. Finally in the case of prediction of time-series, this document only covers models that are of the Finite Impulse Response (FIR) type.

2.2 Model generation

2.2.1 Workflow

The typically recommended workflow for model generation is shown conceptually in Figure 1. First, a series of pre-processing steps are required to select the data relevant for the model generation from the raw dataset. This recommended workflow can be used for both data-driven or first-principle modelling. A short overview of this workflow is given in this section, in the rest of chapter will look at individual aspects of this workflow and their influence on model quality in more detail.

After this, the data is split between a training set (used to fit and validate a model) and a test set (used to perform “external validation”, i.e. to determine expected model quality). The training set is used to fit the model and make choices regarding the model structure. The test set is used to evaluate a model after it has been fitted with the training set. Since the testset was not used when fitting the model, the performance of the model in predicting testset samples will give a good indication of the performance of the model to fit any “new” or “unseen” samples.

Depending on whether multiple model architectures/approaches can be chosen or whether multiple choices can be made for hyper-parameters of the model, a model selection or Hyper-Parameter Optimisation (HPO) loop can be added. In this loop the model is fitted for different choices of the architecture or “hyper-parameters” and N-fold cross-validation is used to assess the expected model quality of each choice and prevent overfitting (see e.g. (G. James, 2017)). Depending on this expected model quality, a choice for the model architecture/hyper-parameters can be made. For machine learning applications, hyper-parameter optimisation (e.g. the number of basis vectors in a Partial Least Squares model, (Wold, 1985)) is commonly a part of the fitting procedure.

For first-principles modelling, this type of optimisation is less automated, but does also occur: the modeller typically changes the model assumptions, architecture and parameters that are estimated based on how well the model is able to represent the data in the training/validation step. . These are typically manual re-modelling steps.

Finally, the chosen model architecture with the fitted parameters is applied in prediction mode to the test set and model quality criteria are applied to judge whether the model is “good enough”. This can be referred to as “External Model Validation”. When the model is deemed not be good enough the entire model generation procedure should be repeated and likely new or additional data needs to be used.

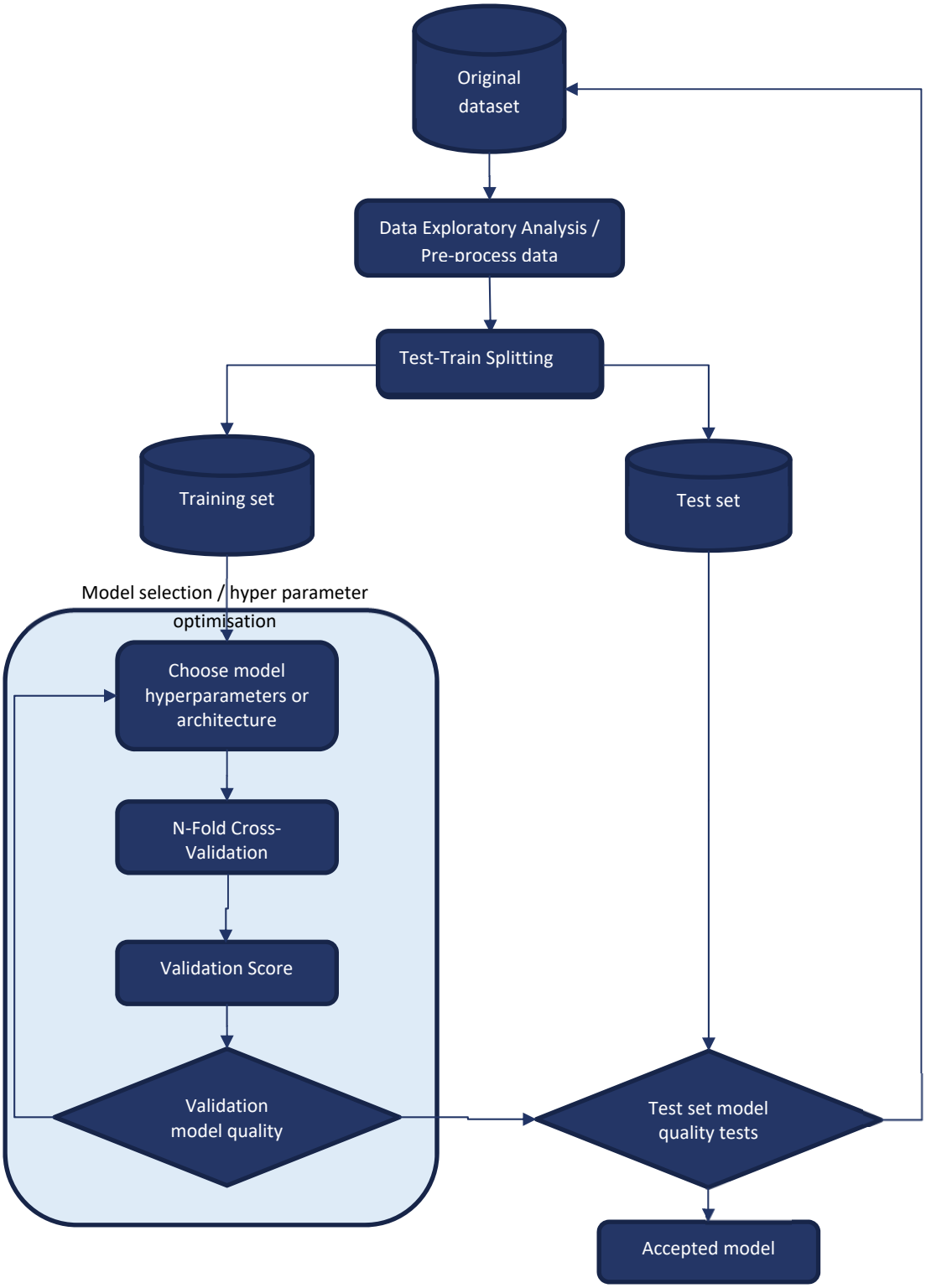


Figure 1 Workflow for model generation

2.2.2 Data-pre-processing

An important step in both first-principles modelling and data-driven modelling is pre-processing of the data. In this step, the raw data that is received is modified in certain ways before it is used as an input to any model validation or machine learning algorithm. These modifications include selection of a subset of the data as being relevant, applying signal processing functions like smoothing, centering, de-trending. Also outliers may be removed and if the data contains heterogeneous quantities, it may be scaled/normalised. These pre-processing steps often involve human analysis and decisions and can account for 90% of the time spent on modelling. During this process, typically also some decisions are made on the "scope" of the modelling endeavour: for which situations should the model give accurate predictions?

2.2.2.1 Selecting relevant data

In many cases, a dataset originating from plant or lab measurements contains both relevant and not-relevant data. Data can be not relevant because of gross measurement errors, errors in the procedure used to obtain the data, or because it covers operating points that are not relevant to the purpose of the model. In these cases it is desirable to remove this data prior to fitting the model.

Before applying any fitting procedures the following algorithms can be used to limit the data that is used for model fitting:

- Selecting data that is in the relevant operating range based on expert knowledge.
- Steady-state detection
- Removal of outliers
- Selecting subsets of data based on clustering algorithms
- General filtering operations

A key point in relation to model quality for these pre-processing steps is that any pre-processing steps that remove certain areas of the original dataset will restrict the validity region of the generated model. For example if only data in a certain operating range is selected, then the generated model will only give valid predictions in this operating range. The effect on model validity of certain pre-processing steps (e.g. restricting the data to the operating region where "high-spec" product is produced) needs to be clearly communicated to the users of the model. Users of a model might, for example, not be aware that a model has been derived from data that has been pre-processed in such a way that the resulting model is not valid under the conditions where they intend to use it.

Process industries

In the process industries the data either originates from measurements and sensors in a laboratory or from instrumentation in-the-field.

Laboratory data in many cases has been generated explicitly for the purposes of constructing the model. It might cover the range of desired conditions well if the experiments have been designed properly. On the other hand, since measurements are expensive, the quantity of data might be limited. Moreover, the scale at which the laboratory experiment has been conducted is different from that of the plant.

When the data originates from online analysers in the field, it is typically obtained from process

historians. These might yield plentiful data from years of operation, but that data might be for a limited number of operating points or it might contain a lot of data that is not of interest (plant startup/shutdown, …).

2.2.2.2 Scaling of input and output data and measurement variance

Before discussing model quality measures, it is important to discuss scaling of input and output data. This scaling process will affect the perceived model quality since model quality criteria are typically applied to scaled output data.

For many applications, the input data contains heterogeneous quantities, (i.e. for example temperatures, pressures and concentrations instead of only temperatures). In addition not all input data has been measured using the same accuracy. Scaling is typically employed to capture this knowledge about the input data in the models that are fitted. Scaling can be done in two ways: per input variable or per (set of) samples. Scaling per variable is often used to account for the fact that different variables in the input data represent different "quantities". In order to assess the relative importance of predicting each of these quantities well, they need to be normalized with respect to each-other in some way.

In machine learning applications, often little a-priori knowledge is available about the accuracy of each particular measurement or it is assumed that quantities can be measured very accurately. For this reason, common scaling approaches are to scale each measurement with its observed variance, range (max – min) or a reference / average value.

In first principles modelling, typically more a-priori information is used about the quality of certain sensors that produce the data. This information is provided for example in the form of sensor noise characteristics. Typical characteristics are constant variance, constant relative variance and heteroscedastic (combination of constant and constant relative variances), with known or estimated parameters. When a constant variance approach is used and the variance is specified as the observed variance in the data, this is equivalent to the typical scaling approach used in machine learning. When a non-constant variance model is employed, the variance of each individual sample depends on the sample value (heteroscedasticity). A first-principles tool like gPROMS (Process Systems Enterprise (PSE) Lmtd., 1997-2018) can also estimate unknown variances together with parameters. This functionality is not employed commonly in practise. It should be used when variances are not known a-priori by the modeller to get meaningful results from statistical tests during model validation (see Section 2.3.2), but it adds extra degrees of freedom the estimation problem which means the problem might be slower and/or more difficult to converge to a solution.

An important difference between first principles modelling and a significant number of statistical modelling approaches is that first principle modelling often produces Multiple-Input-Multiple-Output (MIMO) models where a common set of parameters is used to predict all outputs as well as possible. In statistical learning approaches most commonly a set of Multiple-Input-Single-Output (MISO) models is produced although MIMO models are also encountered. For MISO modelling of data which has multiple measurements, each of the models in this set has its own set of parameters. The implication of this difference in how often MISO versus MIMO modelling is used on scaling is that for first-principle models it is more important to scale the outputs relative to each-other whereas for

most statistical modelling approaches this is less relevant. In practise, this is an advantage of statistical modelling: less a-priori information about the instrumentation is commonly needed.

In certain cases, however, scaling is also important for MISO system. A common example of this is for quantities that can approach or cross 0. Here typically the approach taken depends on whether relative errors or absolute errors are important. In first-principles modelling applications, the error on individual samples is scaled according to the sensor accuracy. This sensor accuracy can be a measure that is relative to the quantity being measured (e.g. "accurate to within 5% of the measured value"). In machine learning applications a log transformation may be applied to the outputs in this case to achieve the same effect.

When there is significant measurement uncertainty for the measured quantities that form one or more of the model inputs, in a first-principles modelling application, model fitting approaches whereby the inputs to the model are assumed to be exact, might be misleading. If model approaches whereby input uncertainty is also taken into account are used, e.g. Total Least Squares (Groen, 1996), then scaling of input data also should be taken into account, because the errors on the inputs become part of the overall criterion to evaluate the model.

Process industries

In the process industries typically, a limited number of physical quantities are measured using online sensors: pressure, temperature, flowrates, composition. In addition, quality parameters can be measured for the final or intermediate products that are often related to material properties (tensile strength, melting point, flow-factor, …). Commonly the variance of each sensor can be inferred from the quantity it measures, the sensor type and/or manufacturer specifications. But collecting this type of information might be a laborious process. In addition, for older assets, this information might not be available.

It was mentioned by COPRO project partner INEOS that raw data from online analysers for steady-states have such a high sampling frequency that measurement noise of sensors is reduced by a large amount since in effect the same quantity is measured a large number of time. This reduces the effective variation on this measured quantity when all the individual measurements are averaged:

$$\sigma_{eff} = \frac{\sigma_i}{\sqrt{n}}$$

Where σ_i is the standard deviation of a given sensor type, n is the number of measurements of that sensor at the same conditions and σ_{eff} is the effective covariance. If n is large the effective covariance of a sensor might be small. For this reason, INEOS commonly does not go through the trouble of obtaining values σ_i but scales data from these sources based on covariances.

2.2.2.3 Centering

Commonly, before applying a fitting procedure for data-driven models, the mean of each variable over all observations is subtracted from the observations of the variable. For first-principle models this procedure is not commonly applied: The reason is that first-principle models generally predict the data including its mean.

2.2.3 Splitting the test and training data

A common procedure in modelling is to split the full dataset in a part used for training (fitting) of the models and a part used for testing or validating the model. This split is typically random, and for steady-state data it is assumed that samples are not correlated. A typical fraction for the ratio of the size of the training and test sets is 70%/30%.

Current approach in process industries

In the process industries, when the data originates from laboratory experiments the quantity of data might be limited. Therefore, in practise, the approach of splitting training and test data is not always followed. The reasons for this is that it is perceived that there is not enough data available to justify leaving a portion of the data out of the training set in order to obtain a statistically sound approach.

For first-principle modelling, an additional reason may be that modellers assume (possibly erroneously) that their proposed modelling architecture has captured the essential behaviour of the process, and therefore the model will extrapolate well. In addition, in place of cross-validation and external validation developers of first-principle models rely on statistical tests based on measurement covariance estimation to prevent overparametrisation.

Finally, for both approaches, if a model is rejected based on the performance on the test set, the approach may be to use the same dataset but change the model architecture and repeat the procedure. In this case, the result may also be a situation where the model architecture and parameters have been influenced by the test set.

2.2.4 Cross-validation

Cross-validation is an approach whereby the available training data is split randomly in a set of samples used for fitting the model (“training set”) and a set of samples for cross-validating the model (“validation set”). The model quality is assessed solely by evaluating the model on validation set based on certain criteria. The purpose of this cross-validation process is to have a criterion to evaluate candidate models on and aid in model selection between these candidate models.

In k-fold cross validations this process is repeated n times, each time selecting n*(k-1/k) samples randomly for the training set and n/k samples for the test set. The validation scores (see next section 2.3 for scoring criteria) from these k-iterations are stored and can be aggregated to yield an overall “validation” score. Its aim is to give an indication how well the model will perform for “unseen” data. The validation score can be used to compare different models and select the best one during a Hyper Parameter Optimisation (HPO) or model selection loop.

After the cross-validation is performed and the validated score is obtained, the model is then re-fitted using the full dataset. The score for this is also stored. This is referred to as the “final” score.

A practical suggestion for the number of folds of the amount of data: generally, 5-fold for small datasets (<100) or 10 fold for larger datasets.

Table 1 Python snippet for performing cross-validation with scikit-learn.

```
from sklearn.model_selection import cross_val_score
scores = cross_val_score(estimator, X, Y, \
    scoring= 'explained_variance', cv=10)
```

Using the cross-validation approach to assess model quality can be problematic when the model is used to forecast a time series. One characteristic of time series data is its inherent serial correlation, so forecasting test data in the direct vicinity of training data may be "too easy" for a machine learning algorithm and hence misleading in controlling the trade-off between overfitting and generalisation capabilities of the model. The common n-fold cross-validation is valid for time-series data in the case of a purely autoregressive model as shown by (C. Bergmeier, 2018).

ClearVu Analytics (Divis Intelligent Solutions GmbH, 2018) offers the user a so-called block-wise cross-validation. With this approach the nfolds are not generated by uniformly choosing data points from the whole data set, but the folds are composed of k time-continuous blocks, with $k \geq 10 \cdot n$. On the borders of these blocks the data is still correlated to its neighbouring blocks. So, to minimise the effect of inherent serial correlation the size of a block should be as large as possible.

Hyndman (C. Bergmeier, 2018) suggests another approach to validate a model for forecasting time series. This approach ensures that no data from the future[1] is used to fit the model. The data is divided into n time-continuous folds. Then the model is fitted with the first fold and tested on the second fold. The third fold is used as test set for a model fitted on the first two folds. So, the amount of training data grows until the nth fold is used as test set for a model fitted on the first n-1 folds.

2.3 Model quality measures

2.3.1 Scoring functions

To define how accurately a model can predict a given set of data samples, a measure of this accuracy can be defined. This is referred to as "scoring". The goal of scoring is to assess whether a model predicts the data well and to assist in model selection (decisions on whether one model is better than another), both when fitting (=training) the data and for cross-validation. It can also be used to convey to the user of a modelling tool quickly whether the model is accurate for a given dataset.

Common criteria for scoring models are listed in

[1] Future w.r.t. the current test point

Table 2, see also e.g. the scikit-learn metrics package ((Pedregosa, 2011), http://scikit-learn.org/stable/modules/model_evaluation.html).

Table 2 Commonly used scoring criteria for regression

Terminology	Abbreviation	Equation	Notes
Prediction Error Sum of Squares	PRESS	$\sum_i^n (y_i - \hat{y}_i)^2$	Also referred to as "residual sum of squares" (RSS) or sum of squared residuals (SRR) or sum of squared errors of prediction (SSE).
Model Absolute Error	MAE	$\frac{1}{n-1}\sum_i^n y_i - \hat{y}_i$	
Mean-Square-Error	MSE	$\frac{1}{n-1}\sum_i^n (y_i - \hat{y}_i)^2$	Also referred to as Mean Squared Prediction Error (MSPE) if applied during cross-validation.
Coefficient of determination.	Q^2 (when applied to test or validation data) R^2 (when applied to training data)	$1 - \frac{\sum_i^n (y_i - \hat{y}_i)^2}{\sum_i^n (y_i - \bar{y})^2}$	$Q^2 = 1 - \frac{PRESS}{TSS}$ Upper bound Q^2<1. Also known informally as "Goodness of fit".
Pearson product-moment correlation	corr	$\frac{\sum_i^n (y_i - \bar{y})(\hat{y}_i - \bar{y})}{\sqrt{\sum_i^n (y_i - \bar{y})^2 \sum_i^n (\hat{y}_i - \bar{y})^2}}$	Used by ClearVu Analytics (Divis Intelligent Solutions GmbH, 2018).
Maximum-Likelihood function	ML-function	$\ln(f) = -\frac{1}{2}nln(2\pi) - nln(\sigma) - \sum_i^n \frac{(y_i - \hat{y}_i)^2}{2{\sigma_i}^2}$	Use by gPROMS (Process Systems Enterprise (PSE) Lmtd., 1997-2018) for model validation.
Chi-squared statistic	${\chi}^2$	$\sum_i^n \frac{(y_i - \hat{y}_i)^2}{{\sigma_i}^2}$	Relies on specification of the measurement variance.
Reduced chi-squared statistic	${\chi_v}^2$	$\frac{1}{n}\sum_i^n \frac{(y_i - \hat{y}_i)^2}{{\sigma_i}^2}$	Also known as Mean Square Weighted Deviation (MSWD)

Here n is the number of samples in the testset, y the measured output data for the tests, $\hat{y}$ the model predictions, $\bar{y}$ the mean of the measurements, σ the user-specified measurement covariance's for each measurement and TSS is the Total Sum of Squares of the test set data, defined as:

$$TSS = \sum_{i}^{n} (y_i - \bar{y})^2$$

In summarizing these criteria for convenience, we assumed a MISO model structure. For the MSE and MAE the Bessel correction is applied to reflect the fact that these provide an estimate of the variance. Additionally, it is assumed that these criteria are applied to the test set. This means that the number of degrees of freedom is n-1. Also, this is restricted specifically to criteria employed to evaluate a chosen model on the test set. Typically, during hyper parameter optimisation / model selection loop criteria that also include the complexity of the model are used (e.g. Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), "adjusted R^{2}") to prevent overfitting at this stage. These criteria might be less computationally expensive to evaluate than an n-fold cross validation, but e.g. a study by Divis (T. Back) has shown output based criteria such as PRESS when combined with cross-validation to be superior in practise for a meta-modelling case study.

Comparing these criteria, the following observations can be made:

- The PRESS error criterion is continuously differentiable and weighs larger errors more heavily. It is not normalized and more difficult to related intuitively to the range variable y itself. The MSE is normalized to a single sample which makes it possible to compare errors on datasets of different sizes.
- The MAE is not continuous but allows errors to be interpreted more easily since this error has the same dimensions as the output itself.
- The Q^2 criterion has the advantage that, because its normalized, it can be used to quickly make an assessment on whether the model predicts most of the variation in the data. For applications where the measurement variance is an order(s) of magnitude smaller than the variance due to changes in the process, e.g. a model with a Q^2 of <0.8 is likely to be inaccurate whereas a model with a Q^2>0.99 is likely to be predictive.
- The Maximum Likelihood function has a foundation in Bayesian inference and is commonly used for parameter estimation problems. If measurement variance values are supplied a-priori, maximising it is equivalent to maximising the (reduced) chi-squared statistic.
- In first-principles modelling or machine learning applications where measurement error covariances have been supplied, reduced chi-squared statistic ($\chi_v{}^2$) fulfils the same role as the Q^2 criterion: it is also normalized and intuitively informative: values close to or below 1 indicate a good model fit (or that the measurement covariance was overestimated…) while $\chi_v{}^2 \gg 1$ indicates a poor fit (or an underestimated covariance).

For applications where the measurement variance is not an order or magnitude smaller than the variance due to changes in the process, Q^2 might not be that informative and it might be better to use the covariance-weighted sum of residuals as an objective criterion: any value relatively close to 1 can be used as a proxy indication of a "good fit".

In addition to the scoring function in

Table 2, two other scoring criteria have been used in COPRO:

1. The Fair function. This function is used in the Surface Condenser modelling case study (UVA, Lenzing). The primary is that it combines a smooth behaviour around error 0 with an absolute error behaviour for larger errors so that outlier are not weighted as heavily as for the MSE:

$$C^2 \sum_{i=1}^{N} \left(\frac{|y_i - \hat{y}_i|}{C} - log\left(1 + \frac{|y_i - \hat{y}_i|}{C}\right)\right)$$

2. The "MSE relative to the mean of the measurements". This is used by COPRO partner TUDO to provide a "normalized" MSE in the absence of covariance information:

$$\frac{(y_i - \hat{y}_i)^2}{{y_i}^2}$$

2.3.2 Statistical tests for model validity

In addition to the scoring criteria, there are a number of statistical tests that can be applied to infer whether the model fulfils certain hypotheses regarding its accuracy with a desired degree of confidence. These statistical tests rely on certain assumptions regarding data and the model (each sample is an independent observation, residuals are distributed normally, linear models). These tests are typically used to aid in a decision on whether to accept a given candidate model. All of these tests rely on the assumption that the measurement covariance has been specified or estimated accurately. All tests are evaluated with a given confidence interval: if a test fails there is a nonzero chance that the test failed due to random variation alone.

Currently all of these tests are implemented as part of the model validation report after a parameter estimation activity in gPROMS ModelBuilder (see (Process Systems Enterprise (PSE) Lmtd., 1997-2018)). In this report we describe the two tests that are most commonly used to evaluate the quality of a candidate model. Additionally the approach of building a learning curve is described, since this yields a practical way of determining whether enough data-samples have been used.

2.3.2.1 Chi squared lack-of-fit test

The chi-squared statistic of the model for the testset can be used to determine whether it is likely that the model represents a good fit. It is assumed to follow a chi-squared distribution with *n-1* degrees of freedom. For a given confidence interval the ${\chi_v}^2$ can be compared to that critical value to determine if the null hypothesis *"The difference between the weighted residual and expected weighted residual is zero"* is satisfied.

2.3.2.2 Parameter confidence intervals

When a model is parameterised with a finite number of parameters, the distribution of for the errors in each parameter can determined from measurement variances and the fitted models as follows:

$$\theta \sim \mathcal{N}\left\langle \theta, \frac{\sum_i^n \left({\sigma_i}^2 \left(\frac{\partial y_i}{\partial \theta} \right)^{-2} \right)}{n} \right\rangle$$

where $\frac{\partial y_i}{\partial \theta}$ is the derivative of a model prediction with respect to the parameter. A student-t test can be applied to distribution to determine confidence internal for this parameter at a given confidence

level. The value is typically compared to the parameter itself to determine whether this parameter has been estimated sufficiently accurately. It is used in particular in first-principles modelling in place of cross-validation since it is an alternative way of determining that the model has been over-parameterized.

2.3.2.3 Learning curve

One question that might arise when evaluating the model quality, in particular when the model quality is deemed to be insufficient, is whether enough data samples are available for fitting a particular model. Inspecting the learning curve might answer this question. The learning curve is generated by taking random subsets of the original dataset with increasing number of samples and fitting the model to those and noting the training MSE and cross-validation MSE. In general, the difference between the validation errors and the training error will reduce the more samples are used for training. Beyond a certain number of samples adding more samples will only marginally improve the cross-validation score. At this point it can be assumed that enough samples have been used to generate and validate the model.

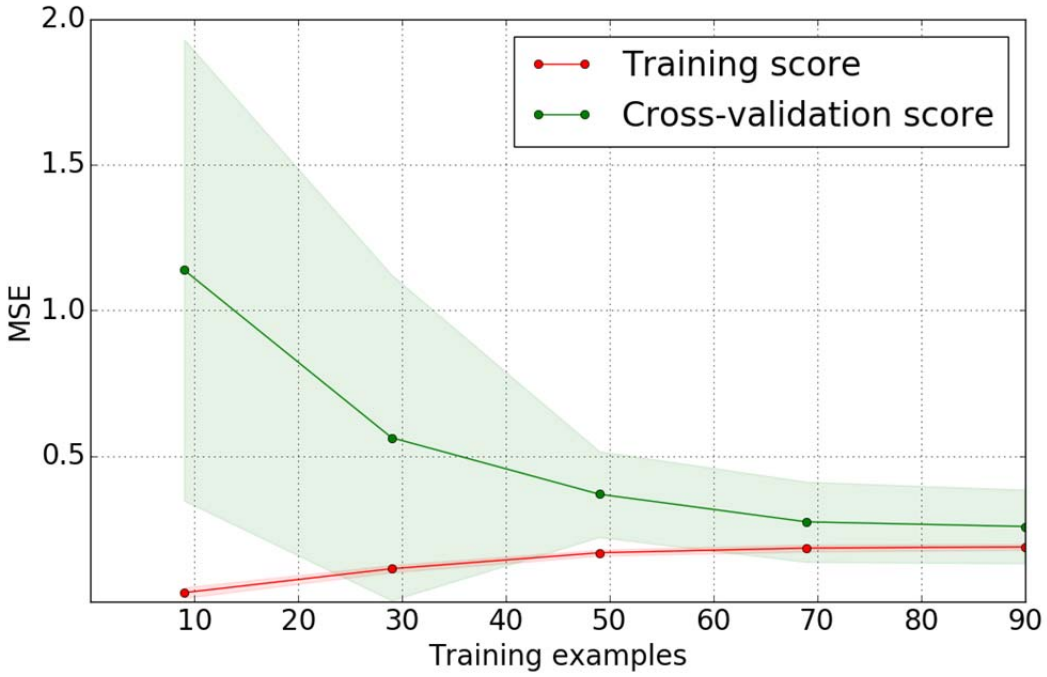


Figure 2 Learning curve for a case study from the pharma domain performed by PSE as part of the COPRO project

2.4 Presenting model quality to the user

The user of a model is interested in generating predictions of the model for his/her own input data set. For this user it is relevant to get the information about model quality that is relevant for this in a quick and concise manner. This user is typically not concerned with reproducing the model generation process, and instead needs an answer to the following two questions:

- (model validity) Under which conditions can I use this model?
- (model quality) How good will the predictions of the model then?

In PSE's Hybrid Modelling tool, the part of the model quality summary information is integrated in the specification dialogs of the data-driven model library. The advantage of this is that the user is presented with this information at the moment linking and configuring in a data-driven model into a flowsheet. This information is just based on the test set used to during the model generation process, it does not contain any information from model evaluations for unseen inputs. The type of information presented is outlined in Table 3 and a screenshot of (an earlier) development version of this dialog is shown in Figure 3. Note that the information presented is for the entire dataset associated with a model, if information with a finer granularity is required (e.g. criteria per subset of experiments) then this summary information is not sufficient.

Table 3 Summary of model quality information as reported to the user of PSE's hybrid modelling tool.

Q^2,R^2	This criterion provides the user with an intuitive measure of the quality of the model in the absence of significant measurement noise. This quantity so can be compared across applications. This allows the user to have an immediate mental association from this number with whether the models are "good" or "bad" for the dataset.	Testset, Cross-validation (Q^2) Trainingset (R^2)
MSE	MSE is commonly used to evaluate models and should be provided for comparison with other tools.	Testset Cross-validation Trainingset
Reduced Chi-Squared	When covariances have been supplied a-priori, this criterion provides an intuitive measure of the quality of the model.	Testset Cross-validation Trainingset
MAE	MAE is commonly used to evaluate models and should be provided for comparison with other tools.	Testset Cross-validation Trainingset

MAE (per output)	This quantity can be used by the model user to make an assessment of how model quality will translate to prediction uncertainty for each individual output. It is a worst-case measure, based on the assumption that the testset prediction error is entirely due to lack-of-fit and that measured error can be neglected.	Testset
Standard deviation of cross validation MSE	This can be used to assess whether the prediction quality of the model assessment is accurate. A large variance compared to the cross validation error criterion can indicate a lack of available data or strong nonlinear effects present in part of the dataset.	Cross-validation
Model validity region upper and lower input limits per variable	Limits per input for the inputs of the model. These input limits can be set by the user during the model generation process or can be determined automatically as the limits for each input in the complete dataset used to derive the model.	Complete data-set
User specified validity region constraints	During the model generation process, the user can define custom constraints to capture restrictions on the validity region of the model. These constraints can be shown in the model quality summary.	User-specified

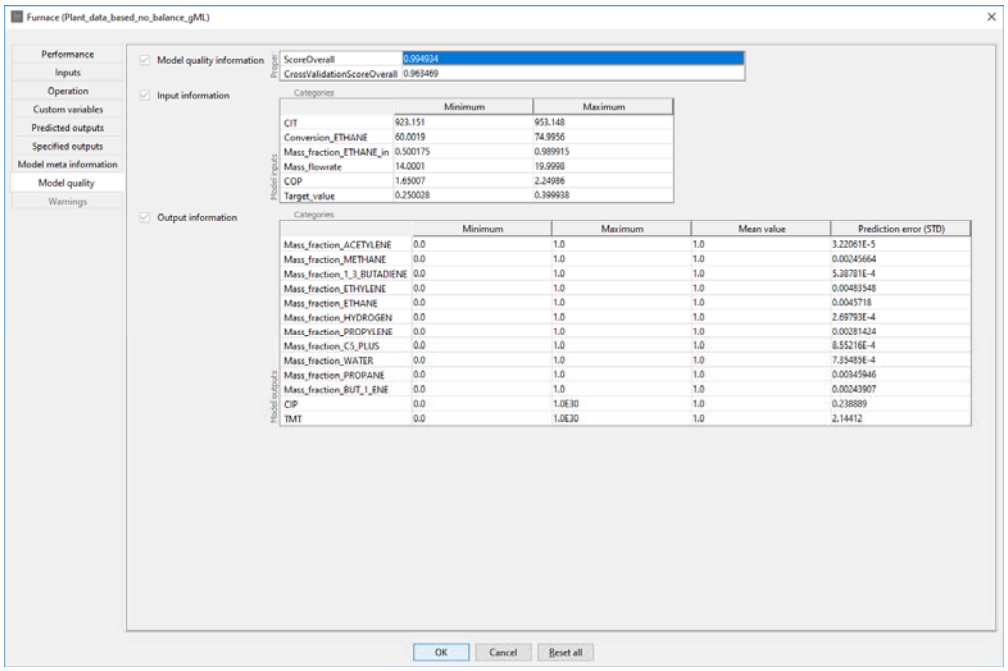


Figure 3 Model quality summary in model specification dialog of PSE's data-driven modelling library

A model quality summary is also presented in ClearVu Analytics at the end of the model generation process. Figure 4 shows a quality measurement table of the three different modelling approaches (multilayer perceptron, linear model, random forest). They are divided into three groups, “Final” being the model fitted on the entire data set, “Learn” are measurements based on the training sets in the cross-validation and “Validation” are measurements derived on the validation sets used during cross-validation. There is a visual indicator (green, orange, red) of model quality which is based on comparison of the “corr” with fixed threshold levels.

In ClearVu Analytics model quality is not monitored specifically for simulations, however for optimisation purposes the user can restrict the optimiser with constraints to enforce model validity for optimised solutions.

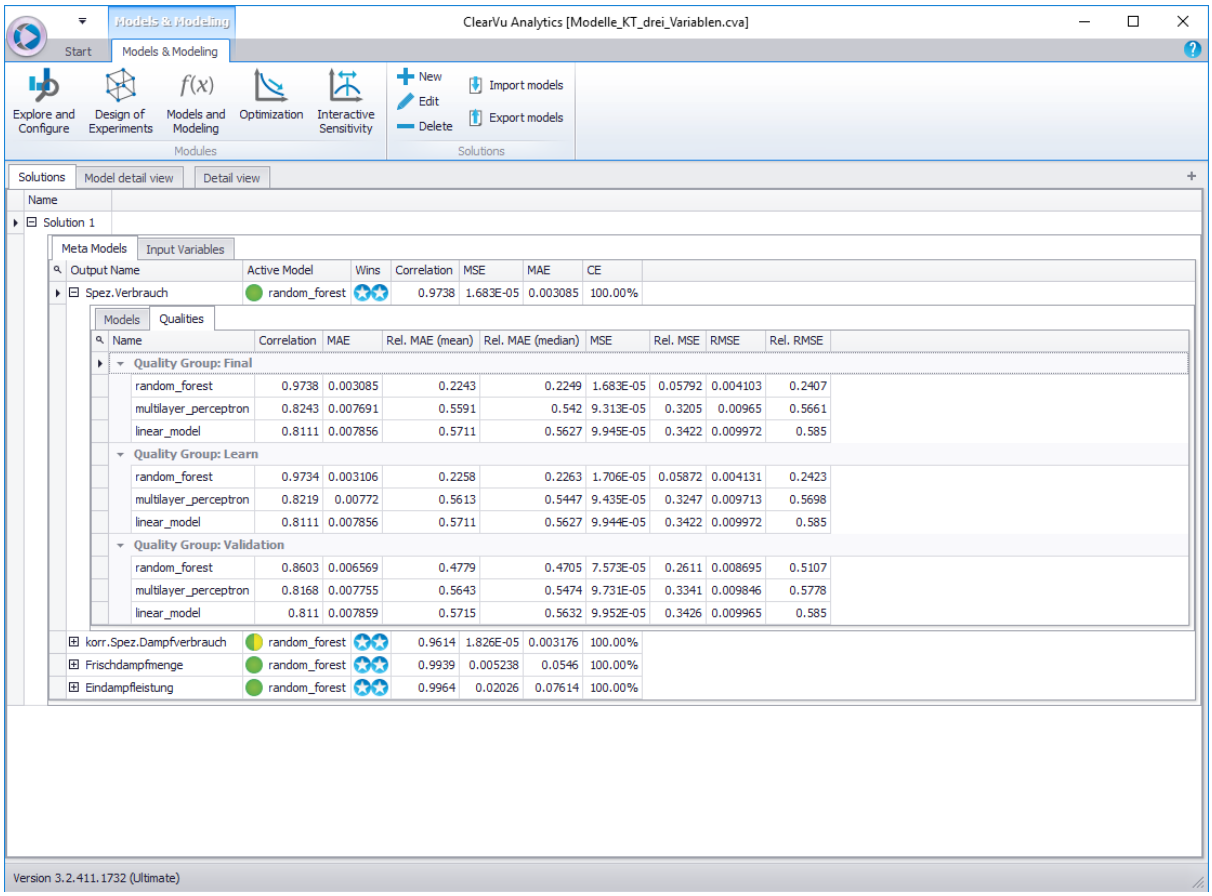


Figure 4 Model quality information as presented in the ClearVu analytics suite.

2.5 Model quality monitoring

2.5.1 Scenarios for monitoring of model quality

When the model is applied to predict outputs from previously unseen inputs it is important to monitor whether these predictions are likely to be accurate and remain accurate over the model life-cycle. In order to arrive at approaches for this, first the different scenarios for model quality monitor must be considered.

With regards to availability of output measurements, we can consider the following different cases:

- Measurements of the outputs are not available (“open-loop prediction”)
- Measurements of the outputs are available (“closed-loop prediction”)

This first case is encountered when the model used to make predictions which are not (immediately) validated. This can be when the model is used in e.g. a modelling tool for a design study, as component of a larger model, etc. For this type of application, it is desirable to have some indication of expected model quality even if outputs cannot be used directly to assess this quality from measurements.

The second case is when output measurements are available. This can occur when a user manually validates the model for a new data-set, or the model is used in online applications where measurements of model outputs are available.

Next, with regards to model quality monitoring, we can also distinguish two situations regarding how many samples are available to make an assessment regarding model quality:

- Model quality is evaluated on a "sample-by-sample" basis
- Model quality is evaluated in a "batch" basis for a given number of samples at the same time.

The first situation occurs when the user of the model performs isolated simulation runs. For each individual run an assessment of model quality is required.

The second situation may occur if a use decides to check the model against a new dataset. It may also happen if the model is used in an online application samples are processes consecutively and a finite horizon window can be used to apply a validation step to the last n seen samples.

For PSE's hybrid-modelling tool developed within the framework of the COPRO project, we focus on the "open-loop prediction" case for the "sample-by-sample" basis.

2.5.2 Model quality monitoring for the "open-loop prediction, sample-by-sample" scenario

If the input data is similar to the data that was used for the training and test sets that were used to derive and to validate a particular model and the process has not changed, the accuracy of the predictions is likely to match the accuracy predicted in the model quality report.

If the input data is not similar to the data used for the training and test sets, the accuracy of the model predictions might be compromised. This similarity between the new input data and the input data used during the derivation of the model can be monitored. When deviations are observed that are deemed to affect the applicability of the model, then the user of the model can be warned.

Apart from monitoring the similarity of the input data to that used for the training and sets by some continuous measure, a more absolute measure of whether in input "valid" can be defined.

When the model is used for optimisation applications, the input space explored by the optimiser can be restricted in such a way that optimisation results will be limited to the input space that is deemed to yield valid predictions.

2.5.2.1 Input validity region

The expected validity of the model predictions based on the similarity based the original dataset and the input space can be captured in certain ways.

First of all, convex regions can be defined in terms of the original model input variables. In the simplest approach the validity region can be restricted to a hypercube defined by limits on each individual inputs and/or outputs. This can be understood easily by users but may over-estimate the true validity region of the model. In particular when the independent variables used in the dataset are highly-correlated and/or represent distinct clusters in the space of independent variables a hypercube might be a misleading representation. An advantage of this approach is that it is simple to understand for the user and also simple for the user during model generation to "override" the individual limits of variables from those in the training sets and test sets during model generation.

A tighter approximation might be the convex hull of the points in the testset. This comes at the cost of a more elaborate calculation and reduced understanding from the point of view of the user.

For models obtained using Partial Least-Squares (PLS) regression (see (Wold, 1985)) there are also statistical tests that measure the similarity between any new input and the reduced input space for the PLS model. Two of these have been implemented in PSE's hybrid modelling tool for model quality monitoring and are shown below.

2.5.2.2 Distance to the model of X-space (Dmodx)

The "Distance to the model of X-space" (DModX) can be used to determine for PLS-type data-driven models whether any new observation is close to the reduced input space (or "X space") of the fitted PLS model (L. Eriksson, 1999). It is defined as:

$$DModX = \sqrt{\frac{\sum_k^m \sum_i^n {e_{ik}}^2}{K - A}}$$

Where K is the number input variables and A the number of principal components in the model and where e are the elements of the matrix E that quantifies the error in the X space:

$$E = X_{new} - TP^T$$

where X_{new} is the matrix with "new" samples (or vector for a single sample), P is the PLS model loadings matrix and T is the PLS model scores matrix.

The DModX can be compared to a critical limit which is found as the inverse of a cumulative F-distribution function for a desired significance level (default = 5%). Warnings can be produced if this level is exceeded.

Note that this monitoring criterion is only valid for "PLS/PCA" type models.

2.5.2.3 Hotelling's T^2

Hotelling's T^2 (see e.g. (L. Eriksson, 1999)) is also a test that can be applied to the input space when PLS models are used. First, based on a new sample the score is calculated, by multiplication with the PLS model loadings:

$$T_{new} = X_{new}P$$

where T_{new} is the matrix with scores for the "new" samples (or vector for a single sample), P is the PLS model loadings matrix and X_{new} is the matrix with "new" samples (or vector for a single sample). These new score(s) are then compared with scores for the testset loading the T^2 distribution:

$$T_i^2 = \sum_{a=1}^{a=A} \left(\frac{t_{new,i,a} - \bar{t}_a}{s_a} \right)^2$$

The Hotelling's T^2 can be compared to a critical limit which is found as the inverse of a cumulative F-distribution function for a desired significance level (default = 5%).

$${T_i^2}_{crit} = \frac{A(N-1)}{N-A} F_{critical}(p)$$

Warnings can be produced if this level is exceeded.

2.5.2.4 Monitoring internal variables and modelling assumptions in first principle models

First-principle models many contain correlations derived from literature. In general, these correlations are derived from controlled experiments and have validity limits. These validity limits are generally specified in terms of bounds on certain physical quantities or non-dimensional numbers (Reynolds, Prandtl, ...). In the same way that model validity is monitored for data-driven simulations. The limits on these variables may also be monitored.

2.5.3 Presenting model quality monitoring information to the user

A key part of model quality monitoring is alerting model users when the model quality is not acceptable. The evaluation of models in PSE hybrid modelling tool is implemented in gPROMS ModelBuilder, which has a flowsheeting capability. Therefore, are warnings on model validity can be displayed directly on the flowsheet by highlighting an icon for a particular model of a unit operation with a red warning outline, see Figure 5. The user can then inspect the model report and determine from there if the model validity region criteria were not satisfied for a particular input or output of if the statistical tests indicate a large dissimilarity between the current input and the testset.

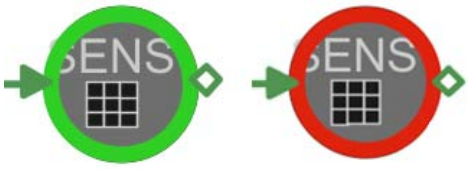


Figure 5 Icons of sensor ("SENS") models on the flowsheet in PSE's hybrid modelling tool turn red when model quality criteria are not met.

2.6 Illustrative examples for model quality and model quality monitoring from the industrial case studies

2.6.1 Industrial Case Study: Granulation (PSE)

PSE has performed a case study (Silva, 2018 (expected)) based on an application from a client: a dry granulation process. In this process powder particles are fed to a roller compactor unit operation where they are compacted into a ribbon. The process ends with the compacting of the granules from the mill in the tablet press. Linked to the output stream of the mill is a Particle Size Distribution (PSD) sensor which measures the particle size distribution. Attached to this PSD sensor is a soft-sensor model that predicts the Flow-Function Coefficient (FFC). This sensor is implemented as a data-driven model using PSE's hybrid modelling tool.

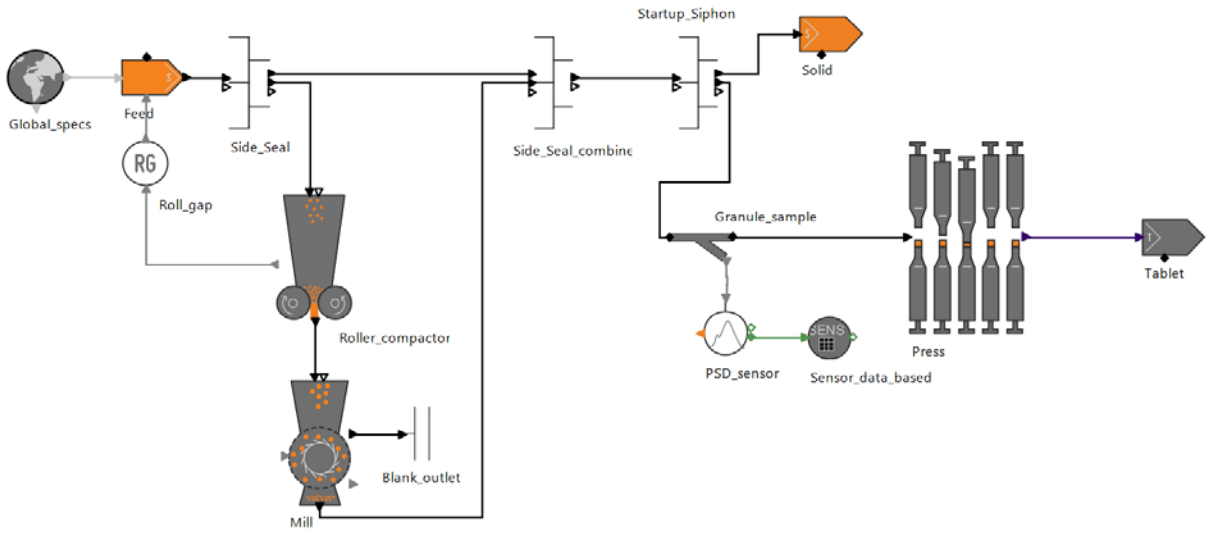


Figure 6 gPROMS flowsheet for the PSE granulation case study

To derive the data-driven model, the relation between PSD sensor measurements (cumulative distributions) and the FFC is estimated from data using PLS regression in scikit-learn. Due to the limited number of measurement samples, a test set split was not conducted and cross-validation results (Q2) were used to determine if a model was sufficiently accurate. Covariances for the FFC measurements were also not available; hence no statistical tests were performed. A model with 3 components was chosen since it maximises the Q2 score. Note that there is a significant difference between the Q2 and R2 score. This is likely due to having a limited number of samples.

Table 4 Cross validation results for the granulation case study

Hyper-parameter (number of components)0	MSPE	Q2	R2
1	0.826	0.174	0.353
2	0.360	0.640	0.735
3	0.351	0.649	0.801
4	0.390	0.610	0.802

The learning curve (see Figure 7) also illustrates this. On the other hand, the gap between training and validation MSE remains stationary from 9-17 samples, indicating that enough samples have are available to capture model behaviour.

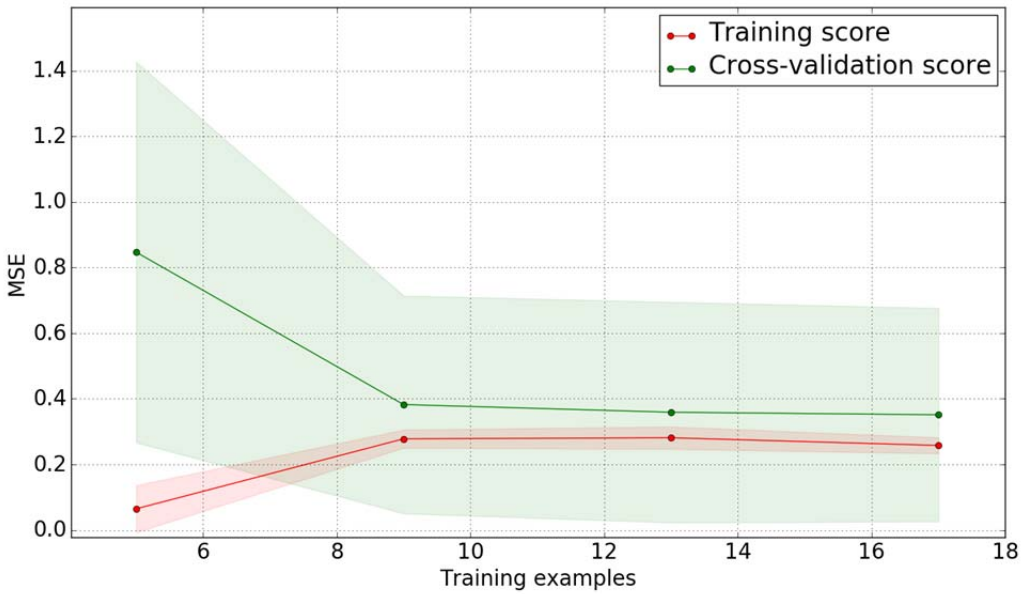


Figure 7 Learning curve for the granulation case study.

When the data-driven soft-sensor model is introduced on the granulation flowsheet and linked to the XML description of the data-driven model, the model quality tab of the specification dialog displays the training score and the cross validation score as well as the model validity limits per individual input and output (see Figure 8).

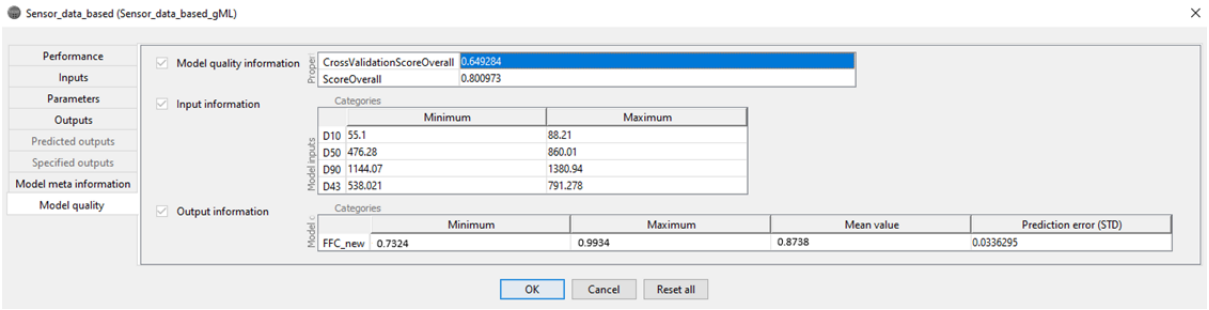


Figure 8 Model quality summary for the granulation soft-sensor data-driven model.

The granulation flowsheet contains unit operations with dynamic holdups. The flowsheet can be solved in a dynamic simulation for variations in key inputs. As the stream conditions change, the PSD measures a dynamic trajectory of particle size distributions. The data-driven soft-sensor consequently predicts a dynamic trajectory for the FFC, see Figure 9. The model quality is monitored using the statistical tests and model validity region specifications. Figure 10 shows that for this particular trajectory, both statistical tests are above their critical levels for much of the time. At these moments the model icon will turn red: the model predictions from this data-driven model cannot be trusted.

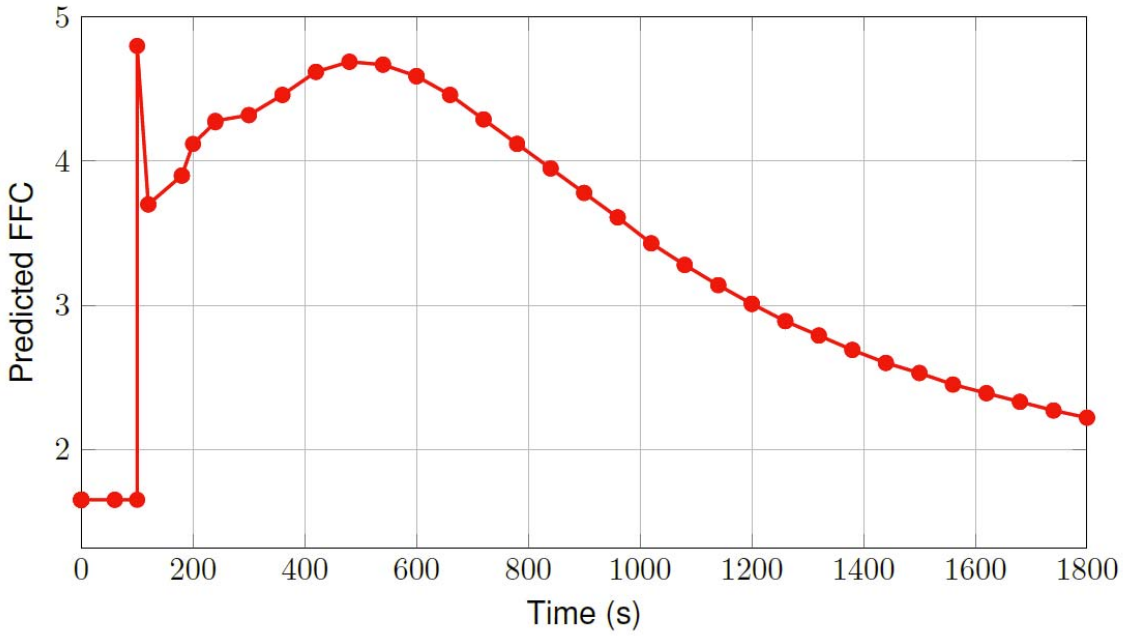


Figure 4.14: Predicted values for the FFC over time

Figure 9 Predicted FFC for variations in time for the granulation flowsheet.

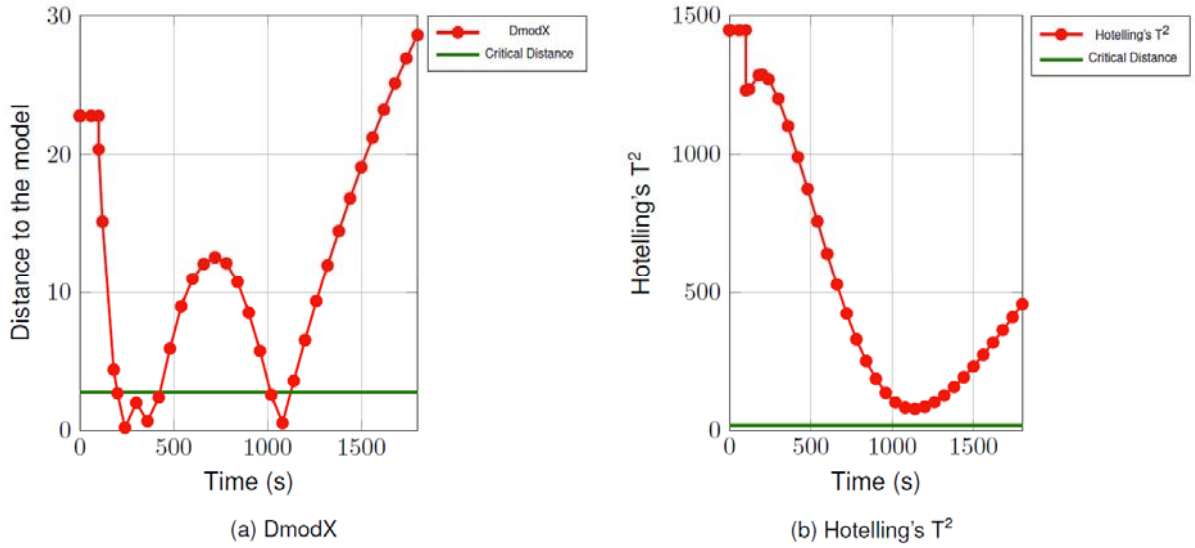


Figure 10 Plots of the DModx and Hotellings T2 statistical tests, along with critical levels, as the granulation flowsheet is solved for a dynamic trajectory.

2.6.2 COPRO Industrial Use Case: Cooling Tower (Divis, Lenzing)

In the Lenzing use case a quantitative description of fouling in the involved equipment is of interest to manage their cleanings. One of the subtasks is to forecast the specific steam consumption of cooling towers involved in the spin bath cycle based on measurements in the past. The data provided by Lenzing covers several months of measurements. The data pre-processing consisted mainly in

defining valid bounds on the measurements and identifying s sections of normal production in cooperation with an engineer on site. This information was used to filter the data. Another step in data pre-processing was to prepare the data set for forecasting, i.e., shifting the output variable (specific steam consumption) by the desired forecasting horizon to the input variables. Then, the data-driven modelling approach of ClearVu Analytics was used with aforementioned block-wise cross-validation to find the best[2] forecasting model. Figure 11 is a scatter plot visualising the forecasting quality of the random forest model. With a perfectly forecasting model all points would be located on the bisecting line. The plot shows the forecasts on the validation sets from the cross-validation in blue and the forecasts of the final[3] model is shown as red points.

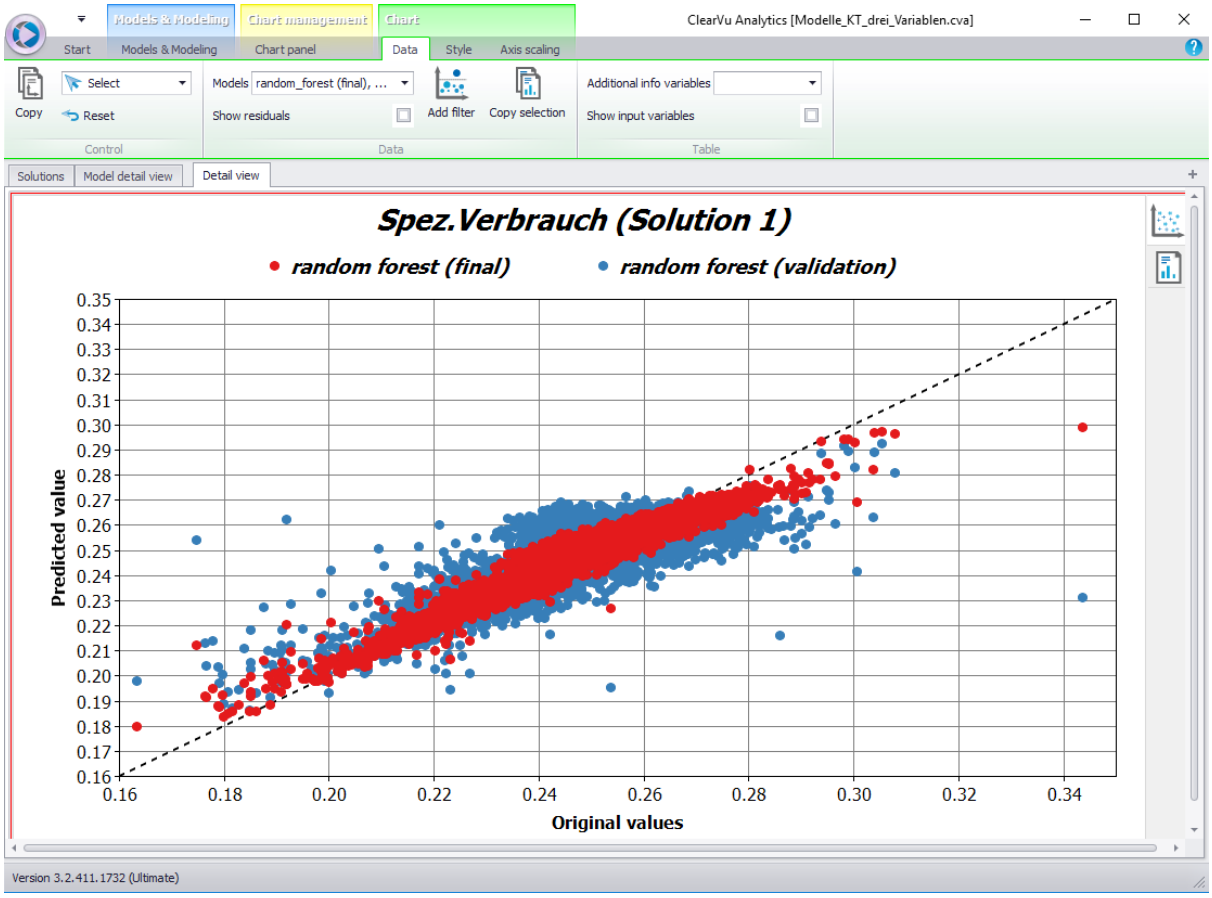


Figure 11 Scatter plot in ClearVu Analytics showing points in the validation and testsets.

2.6.3 COPRO Industrial Use Case: Data-driven models for NH_3 network optimisation (TUDO, INEOS)

Due to the different responsibilities of people that create process models and the various application purposes of such models, the modelling landscape at INEOS in Köln is manifold. In the planning procedure, stationary input-output models are used to predict the resource consumption as well as

[2] With up to 14 machine learning algorithms ClearVu Anylitcs conducts a parameter optimisation and chooses the model as best one which has the smallest MSE on the cross-validation sets.
[3] The model fitted on the entire data set

the raw material uptake rate of the processing plants depending on the desired production rate (DSP models). To use these existing heterogeneous models for optimisation of the ammonia network use case within CoPro, the suitable plant models were translated from the internal and proprietary modelling languages used at INEOS in Köln into a flexible, modular, and open-source framework. This framework, the open source Julia programming environment, comes with integration of various libraries and APIs for integrating data sources, visualizing data, and solving optimisation problems by compiling the mathematical formulation into formats, which are suited for different open-source and commercial solvers.

Additionally, new models were created by TUDO for the remaining plants based on production and planning data provided by INEOS in Köln. These models were created as linear (affine) input-output relations fitted to the provided production data. Although, these models do not capture the inherent nonlinearities of the processes, it has been observed that within the operating windows of the plants, the affine models are capable of predicting the resource and raw material consumption to a satisfactory degree, as will be shown in the following → uniform accuracy for all outputs.

Exemplary for the evaluation of model quality, in Figure 12 predictions are compared to actual data. The red circles denote observed operation of the plants. The three different lines without markers denote the predictions of different origin. The solid line represents the planning data of INEOS in Köln (PLAN), which was taken from the internal planning system on site. The dashed-dotted lines represent the model prediction from the models that were used for planning and optimisation in the FP7 project DYMASOS. The dashed lines show the predictions of the linear (affine) models that were created in the CoPro project. It can be seen that within the operating range, the predictions match the reality most of the time quite accurately. Although, the trend of the data is kept, for some operating ranges there are offsets in the predictions, which could be caused by the nonlinear nature of the true processes. However, these regimes do not dominate the majority of the recorded operating points. The lower part of the figure shows a bar plot with the comparison of the relative mean error of the models when they are compared to the observed operation.

Another assessment of the model quality during the modelling procedure was the computation of the mean squared error between the predictions and the observed data, which is displayed as circle plots for convenience (see Figure 13). It can be seen that for the different streams and this operating regime of the plant, the linear (affine) models outperform the others, including the internally used DSP models. Therefore, these models can be employed in the optimisation of the site without loss of accuracy compared to currently used models.

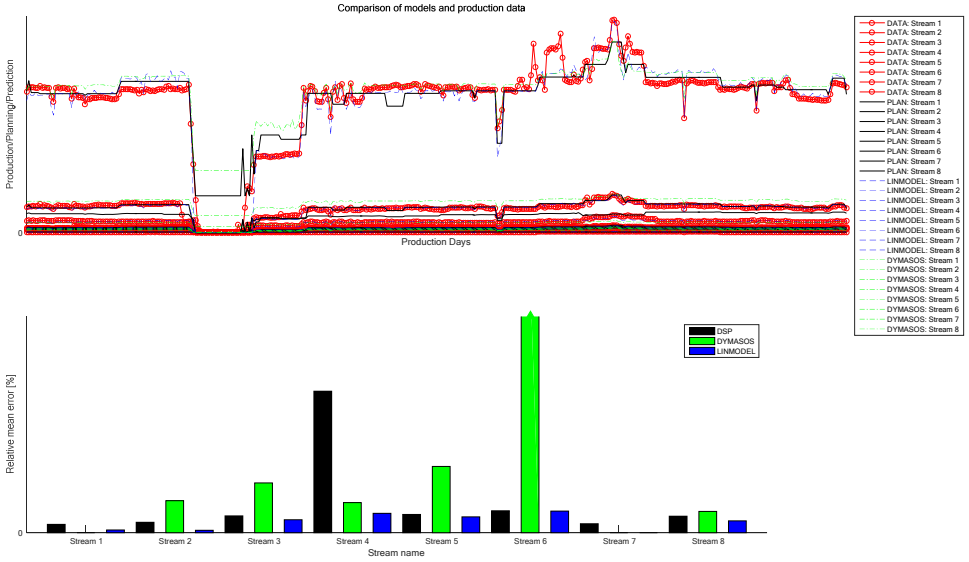


Figure 12: Comparison of different models created for one of the plants at INEOS in Köln.

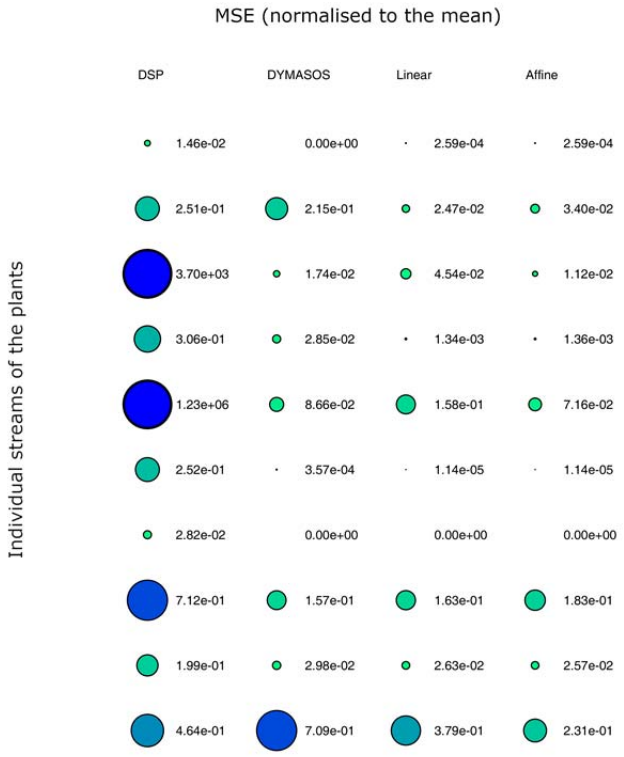


Figure 13: Circle plots for the assessment of prediction quality. On the vertical axis a number of stream for a particular production plant are listed. In the horizontal direction a comparison is possible between the different models that are available. Smaller circles denote smaller mean squared error . This dashboard enables a fast assessment of the quality of the models and indicates the streams with the largest mismatch.

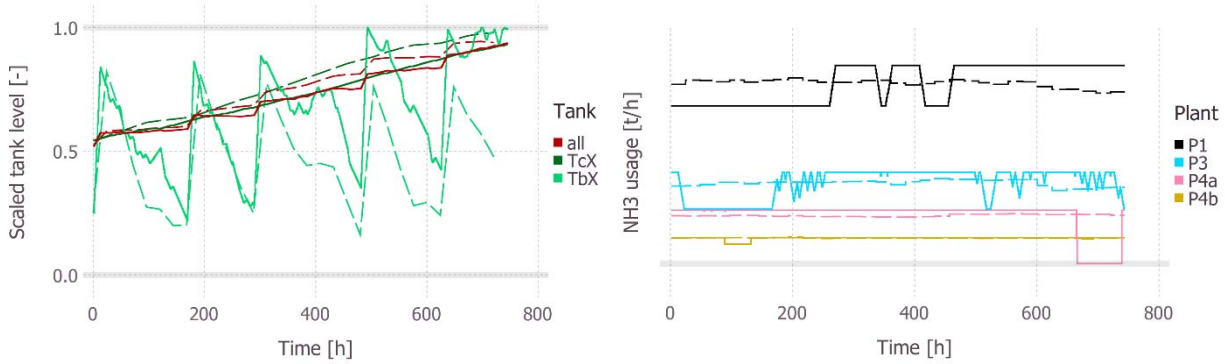
The error is calculated as follows

$$e = \overline{\left(\frac{y_{plant} - y_{model}}{\overline{y_{plant}}}\right)^2}$$

Using these models in the optimisation of the ammonia network to compute an optimal schedule for the operation of the plants, allows for an evaluation of the constraints and a comparison with actual observed operation. Thus, not only the individual plant models but also the overall behaviour, i.e., the relations of the streams between the plants, of the overall model can be evaluated. Exemplarily,

one of these results is shown in Figure 14, where one can see on the left that the scaled tank levels reflect the observed filling and discharging patterns of ammonia at INEOS in Köln. While the pattern structure of measurements (solid) and optimised (dashed) behaviour is similar, the deviation that can be observed when the tanks are being emptied do not result from a mismatch of the models, but they result from the fact that the optimiser chooses a different operational strategy within the constraints in order to save energy for the compression of ammonia (Wenzel et al. 2019). The right side of the figure shows the comparison of the plant operation scheduled by the optimiser, indicated by the dashed lines, in comparison to the observed operation in reality, indicated by the solid lines. It can be seen that the level of production is similar for instance for plant P4b, which is running at a constant production rate most of the time. For the other plants, the different patterns of the trajectories result from the fact that the optimiser assigns different load levels throughout the optimisation horizon. Overall it can be said the model satisfies the linking constraints between the different plants and the accumulated ammonia usage of all plants and matched the reality accurately (Wenzel et al. 2019).

Concluding it can be stated that the models derived from production data of INEOS in Köln serve the purpose within the project and enable a site-wide optimisation of the operating schedule.



a) Scaled tank levels. The dashed lines are the observed tank levels; the solid lines represent the optimised tanks levels.

b) Ammonia usage of the plants. The dashed lines represent the recorded data and the solid lines represent the optimised schedule.

Figure 14 : Comparison of optimisation results with the recorded data at INEOS in Köln.

3 Model uncertainty quantification

Models of a given process never represent the process under consideration perfectly. They are always subject to model uncertainty. This means that any predictions made by a model suffer from a given degree of inaccuracy.

The concept of model uncertainty is closely related to model quality, as model quality criteria aim to capture how accurate a model will predict for unseen inputs. In the Chapter 0 of this report model quality is discussed.

3.1 Model uncertainty quantification in the BDP toolbox (INEOS)

3.1.1 Overview of the BDP toolbox (INEOS)

During the MORE project INEOS in Köln developed a framework for monitoring the resource efficiency of their production plants (Kujanpää, Marjukka and Hakala, Juha and Pajula, Tiina and Beisheim, Benedikt and Krämer, Stefan and Ackerschott, Daniel and Kalliski, Marc and Engell, Sebastian and Enste, Udo and Perez, Jose Luis Pitarch, 2017). The concept of this framework is to provide the operators with a performance reference model named Best Demonstrated Practice (BDP), that represents the most resource efficient and stable production at a specific set of non-influenceable circumstances like ambient conditions or feedstock quality. By comparing the current resource efficiency indicator (REI) with its BDP, operational improvement potentials (OIP) can be identified. Figure 15 depicts an illustrative example that represents the concept. The task of the operator is to keep the OIP, defined as the distance between the REI and the BDP, as small as possible.

During the CoPro project, INEOS in Köln has developed a method to identify a surrogate performance model based on the evaluation of historical process data. It employs an extension of state-of-the-art surrogate modelling techniques, data clustering and model simplification by backward elimination. The details of this approach can be found in (Beisheim, B., Rahimi-Adli, K., Krämer, S., and Engell, S.;, 2018)

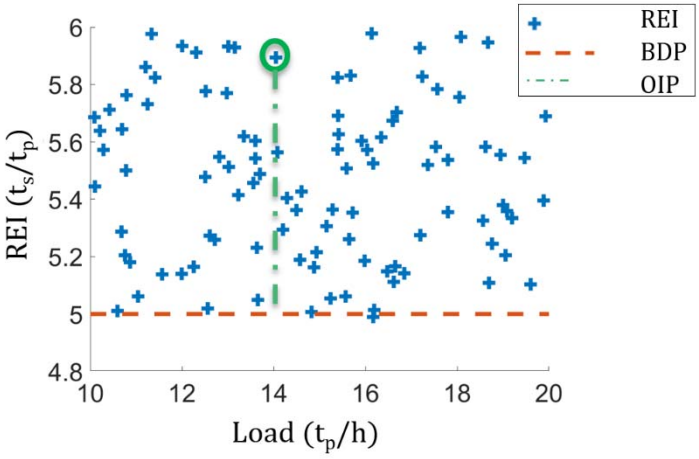


Figure 15 Toy example of BDP concept

3.1.2 Model quality for the BDP toolbox models

In this work package the to the BDP modeling approach is extended with an additional step, which quantifies the probability of the deviation of the resource consumption from the BDP. As mentioned, the BDP model considers those factors that cannot be influenced by the plant operator at the given conditions, e.g. plant load or ambient temperature. However, several other influencing factors/uncertainties are usually present in the process, which influence the resource consumption of the plant. Figure 16 presents an example of a production plant at INEOS in Köln, where the BDP model is shown together with the recorded resource consumption in a given period of time. As depicted, the performance deviates from the BDP. These deviations arise due to the process disturbances and sub-optimal operation, which are related to factors not considered in the BDP model.

A quantification of this performance deviation based on the historical data is advised to estimate the possible resource consumption at given operation points. The approach here is to first calculate the OIP for all of the production data points:

$$\mathrm{OIP}_{r,i} = \mathrm{REI}_{r,i} - \mathrm{BDP}_{r,i} \qquad i = 1,2,\dots, N\ ,$$

where $\mathrm{REI}_{r,i}$ represents the i^{th} data-point for REI of the resource r, and $\mathrm{OIP}_{r,i}$ and $\mathrm{BDP}_{r,i}$ represent the respective values for the same data-point. The next step is done by doing a percentile analysis on the values of OIP and its discretization in to several intervals. The assumption behind this approach is that for an operating condition in the future, the probability of the deviation from the BDP is the same as it has happened for the similar conditions the historical data. Doing so, as shown in Figure 18 instead of having a single BDP model, several resource consumptions scenarios with different probabilities are calculated. Applying this concept to all production plants, the probability distribution of the site steam demand can be computed. This information can be used for a stochastic scheduling of the power plant which is under study at INEOS in Köln. It is expected that the explicit consideration of uncertainties in the site steam demand results in a more efficient resource allocation.

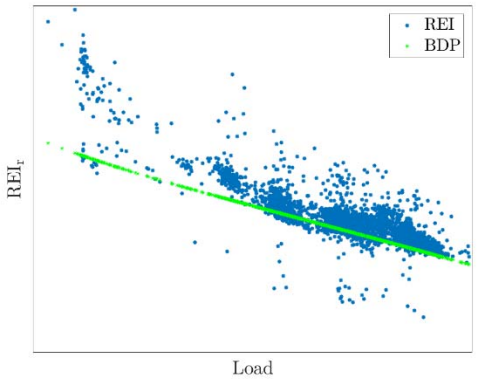


Figure 16 Resource consumption and the BDP model against plant Load

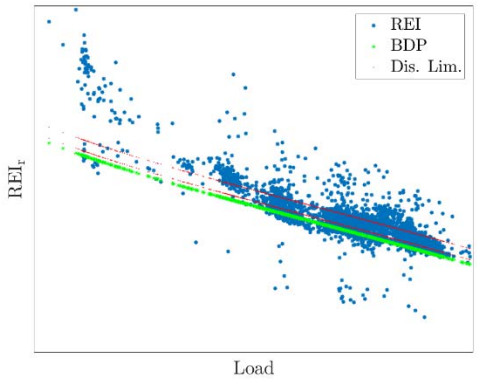


Figure 17 Discretization of the deviations from the BDP model

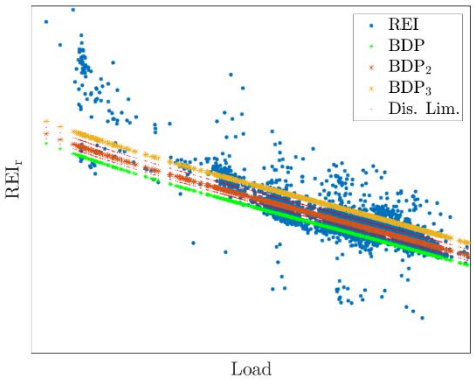


Figure 18 Scenarios with different resource consumptions

3.1 Model uncertainty quantification for PSE's hybrid modelling toolbox

Model uncertainty is considered in a simplified manner in PSE's hybrid modelling toolbox, which is developed within the framework of the COPRO project. During the model generation phase, the Mean Absolute Error (see 2.3.1 Scoring functions is calculated for each individual output of a model for both validation and the testset. The MAE for the testset for is shown, for each individual output, in the model specification dialog (see Figure 19). The user is presented with this dialog whenever the model is first used or configured.

While using the MAE for the individual outputs is informative in most situations, it should be remembered that:

1. This MAE estimate for the testset is worst case because it likely overestimates the true prediction error for a new sample. This is because the recorded prediction error for the testset will include any measurement error in that test set.
2. For a MIMO model the errors in the outputs are not independent. This approach assumes the prediction error for each individual output is not related to that of any of the other outputs.

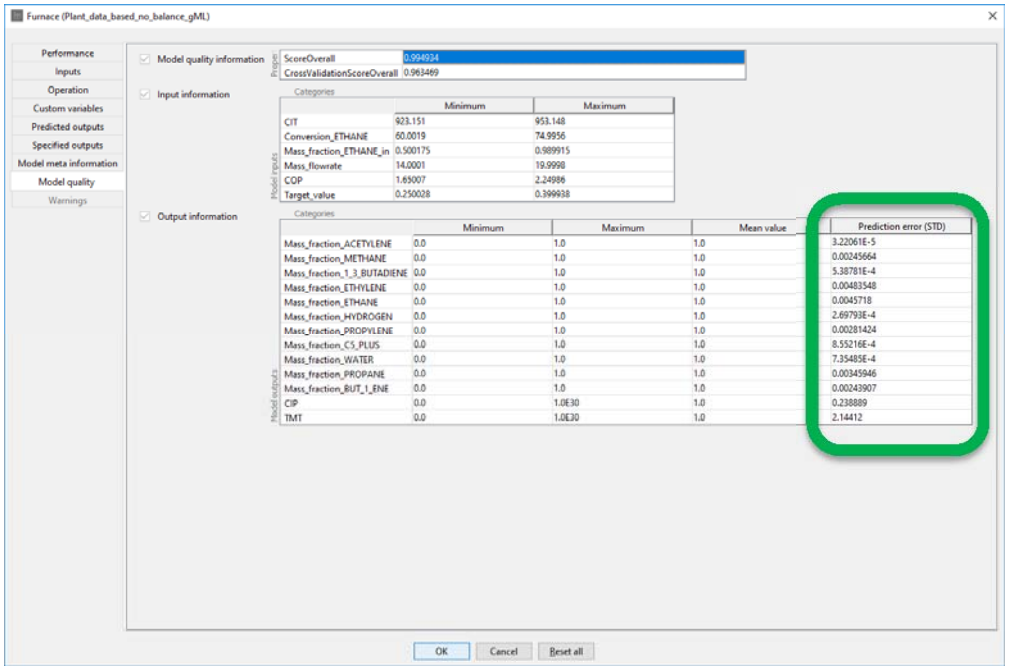


Figure 19 Presenting the user with output uncertainty estimations in PSE's hybrid modelling toolbox. For each output an estimate of the prediction error for unseen data is available based.

For first-principles models PSE's gPROMS platform (Process Systems Enterprise (PSE) Lmtd., 1997-2018) already has existing functionality to quantify model uncertainty. When users have performed a model validation (parameter fitting) task, the confidence interval for these parameters will be determined. This requires a-priori specification or estimation of measurement variances. For a model containing one or more parameters for which the "uncertainty" has been determined from a model validation, gPROMS ModelBuilders "Global System Analysis" functionality can be used to perform a Monte-Carlo analysis of the effects of this parametric uncertainty on model predictions. It will yield a "cloud" of samples that can quantify the combined output uncertainty for multiple outputs.

4 Model maintenance

Maintenance of models in an organization is important in order to maintain the quality of models during their lifetime and promote their use. This involves keeping track of which models are available in an organisation, monitoring the quality of the models and, if the quality is not sufficient, to re-fit or re-model the process. The effort that an organisation is willing to invest in maintaining models may depend on the value of a model to the organisation over its lifetime.

As part of the COPRO project we have considered the following aspects that relate to model maintenance:

- Model accessibility: how easy is it for a user in an organisation to find, use or modify a particular model?
- Model auditing: can someone get insight in exactly how a model was generated? This is important in order to be able to review a model at any point after the time it was generated to decide whether it is still valid or needs to be re-fitted or re-modelled.

The proposed approach for model maintenance for the models developed in the framework of the COPRO project for the INEOS case study is discussed.

4.1 Model accessibility

For models to gain wide use in an organisation it is important that models can be accessed easily. Within the COPRO project this is consider explicitly in deliverable “D5.5 Requirement Specification and Functional Design Specification of the COPRO Model Management Platform“. This proposed management platform allows users to access information about the models within an organisation, access the model meta information and gain access to the location where a model is stored. Accessibility of model also involves aspects such as whether modelling tools can be accessed easily in a corporate IT environment, whether these tools facilitate convenient workflows for predicting and analysing model outputs for new input data and the level of training required for a user to effectively use a particular tool.

4.2 Model auditing

When a model has been generated it is important that its generation process can be reviewed at a later stage. In order to accomplish this, the model generation process must store (serialize) the data used to generate a model, the resulting model itself, and the algorithms used during the generation process in a form that makes the process reproducible. This allows others within the organisation or academic community to inspect and verify results that were obtained. This requires storing the following information (artefacts):

1. Store the original dataset and the associated units of measurement
2. Store pre-processing steps
3. Store test-train-splits for the data or random number generator seed and algorithm
4. Store the scoring methods
5. Store the scoring results and the model selection decisions taken.
6. Store the structure and parameters of the chosen model.

7. Store any "initial guesses" used to initialize the algorithm used to fit the model. This includes initial guess for model parameters and the model architecture.
8. If a "custom" algorithm is used to generate the model, the source code of this algorithm would need to be included. If a "standard" algorithm or tool is used then a reference to the tool and the version can be stored.

Certain modelling tools facilitate workflows where most/all of these artefacts are stored together (e.g. gPROMS case file format (Process Systems Enterprise (PSE) Lmtd., 1997-2018), Divis ClearVu (Divis Intelligent Solutions GmbH, 2018)) while others, in particular those where the workflows are more versatile, (e.g. Matlab (Mathworks, 1994-2018)) rely on the user to explicitly store these artefacts). There are also tools in development to standardise this storage and retrieval process for a wider variety of other tools (e.g. see (DataBricks, 2018)).

A certain degree of standardisation exists in practise for the storage methods of most of the artefacts mentioned above. This type of standardisation depends in most cases on whether the information type by nature is very heterogeneous for each application/model type or whether its homogeneous. For example the dataset that is used to derive the model can commonly be expressed as a (multi-dimensional) table, whereas the description of the model depends on the type of model.

There are different model architectures in statistical modelling and in particular in first-principle modelling. These can be represented in different formats or modelling or programming languages, and a "unified" format generally does not exist. In the Python scikit-learn toolkit (Pedregosa, 2011) for example, the Python interpreter "pickle" functionality can be used to serialize all the objects in the workspace that relate to the model.

Table 5 Types of artefacts of the model generation process and associated storage formats

Type of information	Data format	Standardisation
Dataset	csv, hdf5, xls	High
Pre-processing steps	None	Low
Test-train-split	csv, hdf5, xls	High
Scoring methods	csv, hdf5, xls, xml	High
Scoring results	csv, hdf5, xls, xml	High – limited range of commonly used scoring criteria
Model structure and parameters	csv, (proprietary) modelling software formats, Python "pickle" mechanism	Low – depends on modelling software
Model meta information	xml, json	Medium

4.2.1 Model serialization format

4.2.1.1 Model serialization format for PSE's hybrid modelling tool

For PSE's hybrid modelling tool that is developed as part of the COPRO project, an open file-based model serialisation format is proposed. The approach taken is to define an xml type file format with an associated public schema definition, which is packaged together with the original dataset(s) used to derive the model(s) and any other files associated with artefacts. The xml file stores the information that should permit an independent model audit in a human readable format. The goal is also to make the format tool-agnostic as much as possible. As many of the artefacts are stored in an open/standard format different tools/scripts can be written to import/export this data. Order of tags is preserved and no versioning information is included explicitly to preserve compatibility with source control tools.

Any data pre-processing steps, except selection of samples, are excluded from the format of this hybrid modelling. The reason is that the manipulations done using pre-processing are quite varied and therefore it is difficult to standardise these in a certain format.

The root level of the proposed xml format has the structure as shown in Figure 20.

It contains one or more "DataSources" that represent a source of numerical data. Each DataSource contains either a reference to an external file or has data directly embedded in an XML CDATA tag. DataSources use either csv or hdf5 format. DataSources can contain both, experimental data used to derive the model or data which relates to the model parameters. DataSources expose a map of unique variable names with associated 0 to 2 dimensional floating point data.

It contains tags for "Inputs" and "Outputs" of the model. These contain lists of variables names with associated units of measurements and minima and maxima that form the inputs and outputs of the model(s).

The "DataBasedModel" element has a single property which is the model type. It has two tags inside. The "Parameters" element is a dictionary that maps parameter names (associated with the modeltype) to the variables names exposed by the datatypes. The next is the "VariableTransform" which contains the description of one or more variable transforms that were used for feature generation from the original inputs.

The main tags of interest here are the "ModelQualityInformation" and "ModelMetaInformation" tags. The first contains all information required to present the model quality summary (see "Presenting model quality monitoring information to the user", p.28) in human readable XML format. The second has the model meta information in human readable XML format. This information can be extracted automatically by the model management platform proposed in deliverable "D5.5 Requirement Specification and Functional Design Specification of the COPRO Model Management Platform".

```
<Model>
        <Inputs>
        <Outputs>
        <DataBasedModels type="modeltype">
```

```
                <Parameters>
                <VariableTransforms>
                <ModelQualityInformation>
                <ModelMetaInformation>
        </DataBasedModels>
        <DataSources>
</Model>
```

Figure 20 Structure of proposed xml format for model serialization

4.2.1.2 Serialization for linear/affine models at INEOS

The optimisation model of the ammonia network optimisation will be implemented in a software toolchain that involves the data integration framework of Leikon and the visualisation solutions for an HMI of Sabisu. Hence, the model has to be formulated in a generic fashion that enables the interaction of the HMI with the model. For instance, it has to be possible that an operator manually adjusts a constraint in the HMI which is then propagated through the data integration platform to the model formulation. Hence, in order to ensure full flexibility and efficient handling of the models, only the raw structure of the models, i.e., the constraints and input-output relations, have been implemented within the mathematical models, which are used for the optimisation of the ammonia network. The numerical values of the most recent model parameters, however, are pulled from a database before each optimisation run and constraints with the numerical values are compiled into the solver format. For this purpose, a SQLite database was set up with unique identifiers for each plant, tank, and stream. This data source can then be seamlessly substituted with the data sources that are offered by the data integration platform.

4.3 Model maintenance for the models in the INEOS case study

Similar to the case where constraints can be manually adjusted via the HMI, the model parameters can be adjusted. Whenever a model parameter changes due to various reasons, such as degradation, fouling, cleaning, or bounds on variables, i.e., minimum/maximum tank levels or other process constraints are changed, the database can be updated by the user of the tool via either convenient GUIs that are available to manipulate the database, or the database can be updated by external programs that are linked to production systems via command line queries. Within the envisioned toolchain in the project it would also be able to add an expert interface to the HMI with an expert user account and the respective access rights to change to model parameters. This architecture enables a seamless integration into the tool chain that is planned within the project for a successful implementation of the scheduling tool at the site of INEOS. Major changes to the structure and thus the optimization model formulation are required if for example piping equipment changes or additional equipment is installed. This approach, possibly with a model quality monitoring plugin to trigger the update of model parameters and model structure, will contribute to a concise and clean management of the models during the project and while the tool is run at the industrial partner.

5 Conclusions and recommendations

In this work package the partners in COPRO have examined how to ensure that models can be trusted to predict accurately over their lifetime, how to quantify that degree of accuracy, and how organisations can maintain models.

Central to this is the concept of model quality. This concept, as discussed in Section 2, is established as part of a sound model generation procedure. It quantifies whether models will predict accurately for unseen (or "new") data. Some of scoring criteria used to determine "accuracy" at that point can also be used to get an indication of model uncertainty.

In this section we review the current practise, summarize insights the work on the COPRO industrial case studies performed in the scope of this deliverable has yielded, and give some recommendations to safeguard model quality during the model life-cycle and increase awareness of model uncertainty.

5.1 Conclusions and recommendations related to model quality

There are commonly accepted elements of workflows for model generation, such as cross-validation, splitting of training and testdata and verification of statistical tests (see Section 2.2.1). In first-principles modelling these practises are not followed that commonly, except possibly the verification of a limited number of statistical tests. In the data-driven modelling cross-validation is commonly done and test-train splitting is done in cases where data is plentiful or where the modelling study is organised in such a way that the party involved in modelling does not get access to the test data. Cross-validation is simple to do for steady-state data. For time series data it is a bit more involved due to the care that needs to be taken when splitting the data in "blocks".

Recommendation: The practise of using cross-validation should be followed if only to give an indication of how accurately the model will predict for unseen data. This includes "first-principle modelling" studies where this practise is not yet very common. The model can be fitted using "all the available data" afterwards when data is not plentiful. When data is plentiful it is also recommended to do a test-train split, since this is also easy to implement.

There are a range of measures that quantify how accurately models predict a given data (see Section 2.3.1). These measures are used to summarize model performance for a given set of training, validation and test-data in a single number. They are both used to drive estimation/fitting algorithms as well as to report on model quality at the end of the fitting procedure. In practise, different measures are used in different applications, and typically only a single measure, if any, is reported, making it hard to compare model performance. In machine learning reporting is a bit more extensive as typically at least both the training and validation score are reported in a certain measure.

Recommendation: Modelling tools and users should report a range of scoring criteria after model generation. These are quick to evaluate. This will make comparative studies or quick comparisons easier. These should include criteria that give an intuitive feel for the "absolute" quality of the model (e.g. Q^2, χ^2_v). The commonly used abbreviations for these scoring criteria should appear as labels (MAE, MSE, ...). Due to the potential for confusion between different abbreviations and the "Bessel" and "Non-Bessel" corrected variations a formula should be provided as well where possible. With the possible exception of R^2/Q^2 it is probably better to indicate explicitly to which dataset a criterion

applies (e.g. training/cross validation) than rely on single letter prefixes (e.g. "MSE Training", "MSE Cross-Validation" rather than "MSE", "MSPE").

When scoring criteria are listed directly, this may not be insightful enough to less expert users of a modelling tool. In particular criteria such as "MSE" that are not invariant of the scale of variables being predicted might not give these users a good understanding of the accuracy of the model.

Recommendation: Modelling tools and model management platforms should present model quality in an easily accessible manner for novice users ("traffic light system") both at the end of the model generation procedure and before any use of a model (see e.g. Section 2.4). While this might give irrelevant or subjective information in certain cases, that drawback is outweighed by the fact that this forces all users to consider the model quality aspect of their modelling work. These traffic lights should be derived to Q^2 when no covariances have been supplied or to the cross-validation or testset χ^2_v and its associated "lack-of-fit test" when covariances have been supplied.

Providing a-priori values for covariances of different measurements gives access to a range of statistical tests to evaluate model quality (see Sections 2.2.2.2 and 2.3.2). This is not commonly done for data-driven modelling (machine learning), presumably because certain algorithms do not allow taking this into account and because the data used commonly in the machine learning domain has low measurement uncertainty. In applications where it is difficult or effort-intensive to obtain these covariances they are also taken as the overall signal covariance or for time-series by using frequency-separation (i.e. assume all high-frequency variation is measurement noise). For first-principles modelling this is more commonly done because of the fact that these models are commonly MIMO (see section 2.2.2.2) and because the uncertainty distribution for fitted parameters is an insightful metric.

Recommendation: It is difficult to give a general recommendation on whether it is worth it to obtain a-priori values for measurement variances. This mainly depends on a case-by-case basis on the following factors:

1. *How difficult are the measurement noise variances to obtain?*
2. *What is the relative amount of measurement noise compared to the overall variation in the output signal?*
3. *Is the measurement variance is absolute (i.e. not dependent on each individual measurement value)*

Low effort for obtaining measurement variances as well as high relative amount of measurement noise would motivate a-priori specification. The third factor, whether the noise is absolute, is a pre-condition for easily including the effect of measurement noise variance in some machine learning algorithms, e.g. PLS.

The validity range of a model is often only considered in an ad-hoc manner in the sense that the person using the model has expert knowledge of the limitations of the model. For first-principle models it is sometimes optimistically assumed that the model will predict well beyond the range it has been validated in. Modelling tools generally do not indicate during the use of the model whether the model is valid for the data provided.

Recommendation: Present the validity range of the a model to the user in a way that strikes a balance between being insightful and accurate and indicate using a "traffic light" system whether any "new" data does not fall in this validity range ("model quality monitoring") (Section 2.5.2.1). The person who

creates the model should be able to set the limits of the model validity. The user of the model should be able to adjust the confidence intervals if statistical tests are used to determine the model validity.

5.2 Conclusions and recommendations related to model uncertainty

Currently, model are often provided as-is and users typically do not take model uncertainty into account beyond possibly running a few scenarios based on their understanding ("expert knowledge") of the process and the main uncertainties.

Recommendation: Present a basic intuitive indication of model uncertainty to the user when they are starting to use the model based on the cross-validation or test set score(s) of a model (see Section 3). At least they are then aware of the caveats of using the model. For MISO models this can be the overall MAE and for MIMO models the MAE per output for the cross-validation or testset. This might be conservative in the sense that it overestimates the "true" uncertainty (because the cross-validation or testsets might have an associated measurement accuracy). For MIMO models the MAE per output also might not capture correlations between output errors. But such measure does provide the user with an intuitive indication that might prevent gross misuse of a model. If a more detailed understanding of the input and model uncertainties is required for the purposes of risk assessment studies, the recommendation is to run a Monte-Carlo simulation ("Global System Analysis" analysis in gPROMS).

5.3 Conclusions and recommendations related to model maintenance

Model maintenance is a complex topic that touches on best practises, IT infrastructure, organisational culture and cost/benefit analysis. Within the framework of the COPRO project, not all these aspects can be considered. Instead, the current practise in COPRO partner organisations is considered and some practical recommendations are made to improve best practises with regards to storing results of modelling activities (see Section 4).

Recommendation: First of all, to use proper IT infrastructure where possible to aid in the management on models (version control, access to modelling applications, storage of modelling artefacts). In larger organisations the use of a model management system (see COPRO deliverable "D5.5 Requirement Specification and Functional Design Specification of the COPRO Model Management Platform") can be considered. Next, a recommendation is to store model quality information together with the model in an accessible manner. This model quality information could even be stored at the level of the model management system. An open question with regards to storing model meta information is whether to store this as part of the version control system, within the model file structure or within the model management system. Finally, a recommendation is to use a "snapshot" format for each particular fit of a model. This snapshot format should include the data used, the choices made during the model generation and the resulting iteration of the model and should permit an audit of the model generation process

Bibliography

Beisheim, B., Rahimi-Adli, K., Krämer, S., and Engell, S.;. (2018). Energy performance analysis of continuous processes using surrogate models. *Submitted to Energy, Manuscript No. EGY-D-18-07030*(Manuscript in preparation).

C. Bergmeier, R. H. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics and Data Analysis*, 80-83.

DataBricks. (2018). MLFlow 0.6.0. Von https://www.mlflow.org/docs/latest/index.html abgerufen

Divis Intelligent Solutions GmbH. (2018). ClearVu Analytics. Von http://www.divis-gmbh.de/en/about.html abgerufen

G. James, D. W. (2017). *An introduction to staticial learning.* Springer.

Groen, P. d. (1996). An Introduction to Total Least Squares. *Nieuw Archief voor Wiskunde*, pp 237 -- 253.

Kujanpää, Marjukka and Hakala, Juha and Pajula, Tiina and Beisheim, Benedikt and Krämer, Stefan and Ackerschott, Daniel and Kalliski, Marc and Engell, Sebastian and Enste, Udo and Perez, Jose Luis Pitarch. (2017). Successful Resource Efficiency Indicators for process industries: Step-by-step guidebook. *VTT*.

L. Eriksson, E. J.-W. (1999). *Introduction to Multi- and Megavariate Data Analysis using Projection Methods (PCA and PLS),*. Umetrics. Von https://umetrics.com/products/simca abgerufen

Mathworks. (1994-2018). Matlab Product Family.

Pedregosa, F. e. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 2825-2830.

Process Systems Enterprise (PSE) Lmtd. (1997-2018). gPROMS ModelBuilder.

Silva, A. d. (2018 (expected)). *Hybrid modelling / machine learning for soft-sensing and process modelling.* Lisbon: Lisbon Technical University.

T. Back, P. C. (kein Datum). *Automatic Meta-modelling of CAE Simulation Models.* Divis Intelligent Solutions GmbH.

Wold, H. (1985). Partial least squares. *Encyclopedia of statistical sciences, 6*, pp. 581–591.

Wenzel, S., Misz, Y.-N., Rahimi-Adli, K., Beisheim, B., Gesthuisen, R., & Engell, S. (2019). *An optimization model for site-wide scheduling of coupled production plants with an application to the ammonia network of a petrochemical site*. Optimization and Engineering (submitted).