Checking data informativity as the first step in data-driven modeling – case study

Amir Farzin [a,*], Kateryna Rabchuk [b], Bernt Lie [a], Nils-Olav Skeie [a]

[a] *Department of Electrical Engineering, IT and Cybernetics, University of South-Eastern Norway,*

[b] *Process Modeling and Control Department, Yara International ASA.*

{amir.farzin, bernt.lie, nils-olav.skeie}@usn.com, kateryna.rabchuk@yara.com

Abstract

This paper reviews and introduces the strategies for testing a given dataset sampled from an unknown dynamic process to determine if it is sufficiently informative to model the system's behavior. The presented tests should be done as the first step in data-driven modeling to avoid an endless search for a proper model which may not exist based on the available data. It is unrealistic that available data holds complete information about the system at hand. The tests also allow us to estimate how good the established model can be. Finally, the presented methodologies are applied to an actual process as the case study: modeling the decarbonization section in an ammonia plant.

1. Introduction

By model, we will mean anything for which an experiment can be used to answer questions about the system [1]. Modeling is the act of developing a model. In this definition, the model can be a physical instance of the system or a mathematical representation of the system. The latter is what the model means in this paper.

In many engineering cases, modeling is the first step before other analysis techniques. Therefore, the quality of the model directly affects the solution of the final problem by putting an upper bound on its quality [2]. This fact makes modeling the bottleneck of many engineering problems and raises the need for putting more effort into finding high-quality models.

Data-driven modeling is a rapidly evolving field with great potential to transform engineering science [3]. The concept of data-driven modeling contrasts with physics-based modeling. In the former methodology, the data is the core element that illustrates the behavior and expresses the properties of the regarded phenomenon or object. In data-driven modeling, the scientist does not need to know the underlying physical interactions. These physical interactions form the basis for and needs to be known in physics-based modeling. In the case of data-driven modeling, the relationship between a given set of available measurements (i.e., model input variables or features) and desired behaviors or values (i.e., model output variables or targets) is called the model of the process/system. The construction of a model for a process involves three basic entities:

1. Dataset
2. Model structure
3. Rules for assessing the model from data

The dataset comes first among the items mentioned above, indicating its importance and fundamental effects on the other two entities. The data can be recorded based on designing a proper experiment (e.g., [4]). However, it is not always possible to affect the experiment, and historical data from the plant's operation must be used. Therefore, although the informativity of the data during experiment design is guaranteed, while dealing with a historical dataset, the informativity of data should be evaluated. In addition, the historical data often includes many measurements where only a few are useful for building a model. Therefore, feature selection is a way of reducing the size of the dataset by removing non-informative variables.

This paper presents a method for determining (1) whether the available data is informative enough to develop models and (2) which features contain information about the target variable. The novelty of the paper comes from the informativity of the data. Without considering the fact that the data is informative, one may spend weeks/months to find a model which does not exist. Therefore, at a point, this struggle should be ended by a conclusion that "based on the current data, the process cannot be modeled" or "the data is not enough for the modeling". The current paper helps to make this decision quickly.

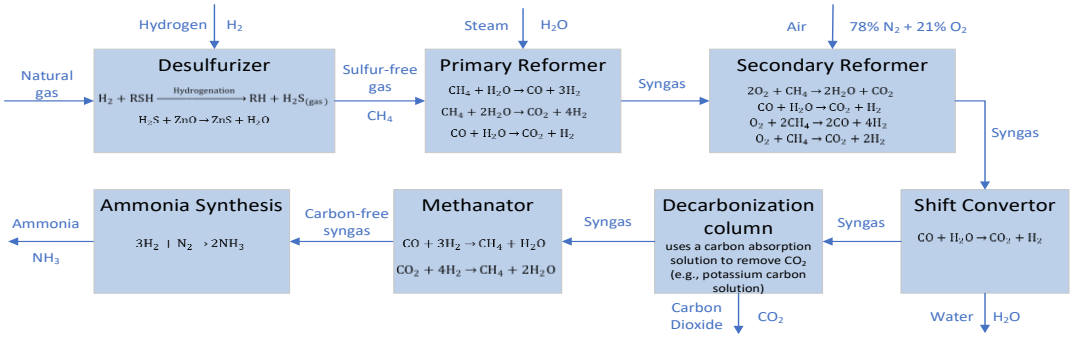

Figure 1: Overview of ammonia typical production steps

The viewpoint of this paper is more practical than theoretical, and represents a walkthrough on how to start modeling given a dataset without any further knowledge about the process. A case study from an ammonia plant is used to illustrate the methods. The case study is an example of non-informative dataset which serves to illustrate the topic of this paper.

In the sequel, Section 2 gives a brief presentation of the dataset. Section 3 reviews the tests for checking the informativity of the dataset and selecting the features. For each represented test, the result of applying it to the case study dataset is shown with further discussion on the results. The authors of the present paper suggest to apply the proposed methods to the dataset in the suggested order, and if applicable, the latest results from each step are used as the basis in the subsequent steps. Section 4 provides a discussion on the results and the conclusion is given in Section 5.

2. Case study dataset

Yara International ASA, one of the largest ammonia producers globally [5], has provided the historical data used in this work. The data is coming from a conventional steam reforming ammonia plant with natural gas used as both feed and fuel. The production capacity is approximately 400,000 tons/year.

As mentioned above, the assumption in the following approaches is that no technical information about the data is available. In other words, it does not matter whether the data comes from an ammonia plant, a refinery, or any other industrial process. However, as the essence of the features and target is known, it is helpful for evaluation purposes. Therefore, let us briefly explain the process and data for a more straightforward interpretation of the results.

2.1. Ammonia plant process

Ammonia plants normally use natural gas to produce ammonia. In the process, natural gas is converted into hydrogen, and then the hydrogen is combined with nitrogen to produce ammonia using the Haber-Bosch process [6]. Figure 1 illustrates the whole process with the main chemical reactions in each block. The process is described briefly in the following.

Natural gas contains hydrogen sulfide components that can deactivate the catalysts used in the further steps of the process. Therefore, the first step is to remove sulfur components from the inlet gas; the so-called desulfurization. The sulfur-free natural gas is then sent to the primary reformer to react with super-heated steam, where H_2, CO, and CO_2 are the products. Then, the gas, called synthesis gas or syngas, is mixed with air in the second reformer. The air's nitrogen is needed in the final synthesis, and the oxygen reacts with syngas to produce more hydrogen. To convert the CO contents of the syngas into CO_2, it is sent to the shift converter section. At this point, the residual water is also removed from the syngas. Then, in the decarbonization section, CO_2 is absorbed from the syngas. Next, the outlet of the decarbonization section passes through another purification section called the methanator to remove small traces of residual CO/CO_2 from the syngas by converting them to methane. Finally, to produce ammonia, the purified syngas (which now contains almost only H_2 and N_2) enters the ammonia converter or synthesis section, where the ammonia is the final product.

2.2. Target variable: CO_2 slip

In the methanator, the reaction between CO/CO_2 and H_2 is extremely exothermic. Therefore, high amounts of CO/CO_2 supplied to the methanator can increase the reactor temperature leading to a series of complex reactions where the consequence is temperature runaway [7]. In addition, removing higher amounts of CO_2 in the methanator consumes more power and reduces the efficiency of the plant. Therefore, it is crucial to control the CO_2 residual in the syngas that enters the methanator (/leave the decarbonization column) below a desired limit. This value is the so-called CO_2 slip, the target variable for modeling in the case study dataset.

2.3. Features

Based on experts' knowledge[1], 45 values from historical data are provided for the modeling. Although the selected variables are initially assumed to be promising, there is no guarantee whether they carry sufficient information about the target. In addition, there is a possibility that some variables are redundant, which means they have almost identical information. Using two redundant features does not help the modeling process and may make the model problematic by wrongly assuming the redundant information too important in the training phase. Interestingly, this is the case regarding the dataset used here.

The values of 39 features come from 13 controllers (C1-C13). For each controller, three measurements are available: (1) set-point (SP), (2) sub-process output measurement (PV), and (3) control signal (OUT). If the controller is doing its job well (the case in most running plants), the PV value tracks SP, which means they are almost identical. In some cases, if a simple PID controller is used, the OUT value is nearly proportional to the PV values. However, this is not always true because of non-linearity and saturation, or in the case of cascade controllers. All 13 controllers with a brief explanation are listed in Table 1.

Table 1: List of controllers in the dataset

Code	Type	SP/PV	OUT
C1	Flow	Steam flow (plant load)	Valve
C2	Ratio	Steam-to-carbon ratio	Ctrl (gas flow)
C3	Ratio	Gas-to-air ratio	Ctrl (air flow)
C4	Temp.	Prim. reformer temp. (last 1/3, near outlet)	Ctrl (pressure of last burner nozzle)
C5	Temp.	Prim. reformer temp. (first 1/3, near inlet)	Ctrl (pressure of first burner nozzle)
C6	Δpressure	between air and syngas inlets, sec. reformer	Valve
C7	Flow	Semi-lean solution	Valve
C8	Flow	Semi-lean solution	Valve
C9	Flow	Purge gas into unit	Valve
C10	Flow	Purge gas into prim. reformer	Valve
C11	Pressure	Syngas compressor suction	Ctrl (compressor)
C12	Flow	Lean solution	Valve
C13	Flow	Lean solution	Valve

The remaining six variables (S1 to S6) are measured signals from sensors, where five of them are temperature sensors, and one is a gas sensor. Table 2 summarizes the features, including the responsibilities of controllers and sensors.

Table 2: List of sensors in the dataset

Code	Type	Description
S1	Temp.	Cooling water temperature
S2	Temp.	Ambient temperature
S3	Temp.	Secondary reformer catalyst temperature
S4	Temp.	Secondary reformer outlet temperature
S5	Gas	Secondary reformer methane slip
S6	Temp.	Syngas temperature exits shift convertor

[1] Engineers at Yara International ASA

2.4. Samples

The process historical data is stored in one-minute intervals. The dataset consists of data from 30 consecutive days in August and September 2020. Therefore, there are 43200 samples, each having 46 measurement values (i.e., 45 features + target). Note that, for secrecy, the actual values of the variables are normalized into range [0,1].

3. Step-by-step methodology

Assume a dataset from an unknown industrial process is given, and the desired output (i.e., target) is pre-defined. Therefore, the rest of the variables are inputs to the model (i.e., features). In this section, a practical walkthrough for the starting phase of modeling is presented to check the informativity of data and select the useful features.

The walkthrough consists of three main steps: (1) data visualization, (2) data splitting and initial modeling, and (3) feature selection. The feature selection itself, is divided into two steps.

Note that, for the computations, data science packages in Python such as NumPy, Pandas, scikit-learn, Seaborn, etc., are used.

3.1. Data visualization

Data visualization is data representation by employing visual elements like charts, graphs, and maps, which helps to comprehend the trends, outliers, and patterns in data. Therefore, it is recommended to start data-driven modeling by data visualization prior to the other steps.

As we are dealing with time-series data, the first essential plot is a simple time plot of each variable. To save space, only the target variable and C9_SP are shown in Figure 2. In addition, the weekends are highlighted in the plot, which sometimes helps to realize if they have different trends.

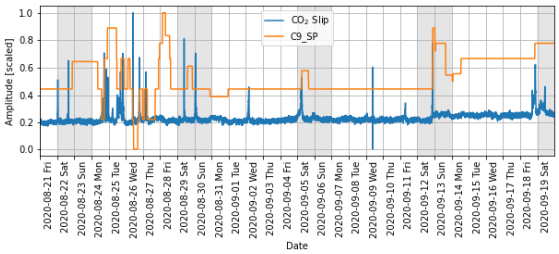

Figure 2: Time-series plot of the target variable (CO_2 Slip) and one feature (C9_SP)

In this example plot, one obvious outlier in target data can be seen on Sep. 9. Another visible issue is a rise in the mean values of both variables after Sep. 12, which suggests the usefulness of C9_SP for modeling CO_2 slip. Note that this relevancy is not the case for all features. Several sharp peaks in

CO_2 slip values indicate a skewed probability distribution of this variable.

Suggestion: It is beneficial to plot the time-series data using interactive plots. Python, for instance, has rich libraries for interactive plots where the programmer can plot all features and targets together. Then it is possible to switch on/off any variable or zoom in/out to have better comparison and inspection.

Another helpful visualization technique is the histogram plot which reveals the data distribution. For instance, Figure 3 depicts the histogram for CO_2 slip where the data bins before and after Sep. 12 are shown in different colors. In addition, the skewness of the data during the whole period is also shown, which is a relatively high number.

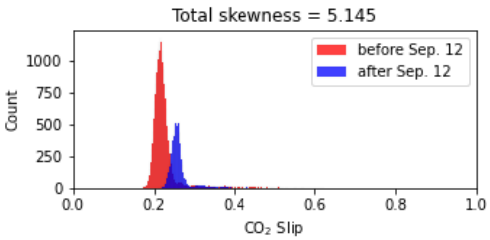

Figure 3: Histogram plot for the target variable

Suggestion: Most linear system identification methods (e.g., ARX, FIR, etc.) assume normally distributed data and Gaussian noises [8]. In cases similar to this example, it may be beneficial to try different models for each part. Later, using a clustering method can define which model should be used in each situation.

Several other charts and plots (e.g., bar chart, radar chart, etc.) can give an initial insight into the data. However, for saving space, this is skipped here.

3.2. Initial regression and dataset split

Before examining the features and selecting the promising ones, making an initial regression model is beneficial. This model reveals if the data has the potential to predict the target variable. In addition, the regression results can help for feature selection later.

In data-driven modeling, splitting data to train and test sets is a routine. However, instead of breaking the dataset into two simple parts, let us divide it into four parts with the following breaking dates: Aug. 28, Sep. 5, and Sep. 12. Then, four train-test pairs are made by taking each part as the test set and the rest as the train set (Table 3).

Table 3: Four train-test pairs

Index	Train	Test
1	Part 1 + Part 2 + Part 3	Part 4
2	Part 1 + Part 2 + Part 4	Part 3
3	Part 1 + Part 3 + Part 4	Part 2
4	Part 2 + Part 3 + Part 4	Part 1

* Part 1: Aug. 21~Aug. 28, part 2: Aug. 29 ~ Sep. 5, part 3: Sep. 6 ~ Sep. 12, part 4: Sep. 13 ~ Sep. 19

The idea comes from k-fold cross-validation [4]. Here, one may be concerned about the test sets' validity as the system is dynamic and the prediction needs the data to be continuous in time. However, as each train set has more than 30,000 samples, and typically not more than 100 past values are used in the models, this will not cause any problem.

The form of the ordinary linear regression (OLS) model for a dataset with n features and N samples is as follows:

$$y(k) = \theta_0 + \sum_{i=1}^{n} x_i(k)\theta_i \tag{1}$$

Using the extended feature vector $X(k) = [1, x_1(k), x_2(k), \dots, x_n(k)]$ allows writing Eq. (1) in the following compact form:

$$y(k) = X(k)\theta \tag{2}$$

where $\theta = [\theta_0, \theta_1, \dots, \theta_n]^T$ is the coefficient vector. The coefficient vector θ can be calculated by minimizing the mean square error (MSE) between model prediction and measured values:

$$\theta = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T Y \tag{3}$$

Here, $\boldsymbol{X}_{N\times n}$ is the feature matrix where each row presents one measured sample, and $Y_{N\times 1}$ is a column vector containing values of the target.

While dealing with dynamic systems, it is beneficial to use the finite impulse response (FIR) instead of a simple (static) OLS. From one point of view, FIR is the dynamic version of static OLS where shifted values of the features are added as new features to the model. Therefore, the FIR model has the following structure:

$$y(k) = \theta_0 + \sum_{i=1}^{n}\sum_{j=0}^{m} x_i(k-j)\theta_{ij} \tag{4}$$

where m is called the order of the model and represents the maximum backward shifts for x. To calculate the coefficients θ_{ij}, Eq. (3) can still be used. However, the coefficient vector θ, feature matrix $\boldsymbol{X}$, and target vector Y should be constructed as follows:

$$\theta_{1\times(\mathrm{nm}+1)} = [\theta_{01}, \theta_{02}, \dots, \theta_{0m}, \dots, \theta_{n1}, \dots \theta_{nm}]^T \tag{5}$$

$$\boldsymbol{X}_{(\mathrm{N}-\mathrm{m})\times(\mathrm{nm}+1)} = [X_1^{(0)}, X_1^{(1)}, \dots, X_1^{(m)}, \dots, X_n^{(0)}, \dots, X_n^{(m)}] \tag{6}$$

$$Y_{1\times(\mathrm{N}-\mathrm{m})} = [y(m+1), y(m+2), \dots, y(N)]^T \tag{7}$$

where:

$$X_i^{(j)} = [x_i(m+1-j), x_i(m+2-j), \dots, x_i(N-j)]^T \quad (8)$$

for $i = 1, \dots, n$ and $j = 0, \dots, m$.

The advantage of FIR compared to static OLS is that the past values of features affect the current target prediction. Therefore, if some features affect the target with delay, FIR model includes those delays in the correspondence coefficients.

Now, let us build three models for each train-test pair. The first model is a static OLS, the second and the third are FIR with orders 10 and 30, respectively.

As we want to make a linear model for the process, the dataset is informative enough if the residual values have zero mean. Therefore, for each model, the distributions of residuals in the prediction of test sets are plotted in Figure 4.

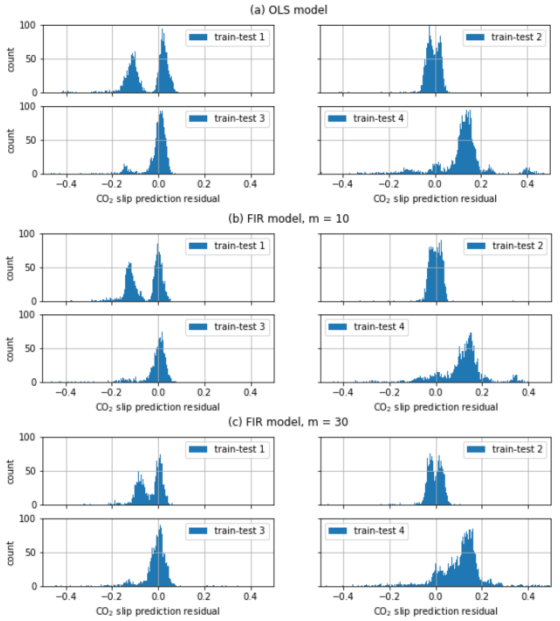

Figure 4: CO_2 slip prediction residual using in (a) OLS model, (b) FIR model m=10, FIR model m=30

The above plots show that the trends in the first part (i.e., train-test 4) and the last part (i.e., train-test 1) of the data are not wholly similar to the other parts. However, using a FIR model of higher order makes it possible to predict the mean value change discussed earlier, which occurred on Sep. 12. In contrast, for the first part of data, it is almost impossible to accurately predict the target values using either FIR or static OLS models.

As mentioned earlier, redundant or irrelevant features can also cause the above problem. Therefore, let us examine the features and select the promising ones in the following subsections.

3.3. Feature selection: filter methods

Up to this point, we have some idea about the dataset. However, we need some metrics to select valuable features. Generally speaking, feature selection methods are classified into three major groups [9]:

1. Filter methods
2. Wrapper methods
3. Embedded methods

Going through all feature selection methods is out of the scope of this part. Instead, a few filter methods are explained and applied to the data in this subsection. The next subsection briefly reviews the wrapper methods. The embedded methods are not discussed in this paper.

Filter methods are simple statistical tests independent of the model structure. They are computationally cheap and easy to apply to big datasets. Therefore, it is suggested to use them before other methods.

The first and most important filter method is the Pearson correlation coefficient test which measures the linear relationship between variable pairs. The correlation of two random variables is the division of the covariance by their standard deviations:

$$\text{corr}(A,B) = \frac{\text{cov}(A,B)}{\sigma_A \sigma_B} \quad (9)$$

The correlation values are in the range [-1,1] (if the variances of both variables are non-zero[2]), where the value of 1/-1 means that the variables are positively/negatively proportional. Several modeling methods use linear regression as the assessment rule. Hence, a feature that has a near-zero correlation with the target variable almost contains no information about the target.

In addition, if two features are highly correlated (e.g., |corr|>0.95), they almost contain identical information, making one of them redundant.

The correlation matrix is a symmetric matrix that presents the correlation values of each variable pair. For the case study, the matrix is 46 by 46 and is massive to show here. However, the matrix for the first 24 variables is shown in Figure 5 using a heatmap. The correlations between SP and PV values for all controllers are almost 1.0. This was expected, as discussed earlier. Therefore, from each pair, one of them is redundant. In addition, SP/PV values of controllers C4 and C5 are also correlated. C4 and C5 control the temperature of the last 1/3 and first 1/3 of the primary reformer, respectively. Therefore, it is logical that both temperatures are kept proportional. The same thing is true about controller pairs C7-C8 and C12-C13. C7 and C8 are two parallel controllers for the semi-lean solution, and C12 and C13 are two parallel

[2] A random variable with zero variance has no information and can be removed from the dataset.

controllers for the lean solution. Therefore, identical set-point settings and their correlations make sense.

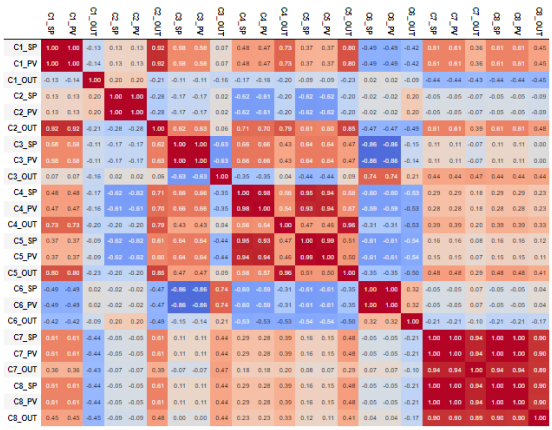

Figure 5: Correlation matrix of the first 24 features

Also, as expected, some OUT values have high correlations with their corresponding SP/PV values. However, we keep all of them.

A correlation test can also be done between the target and each feature. Figure 6 depicts the correlations between the target and all features.

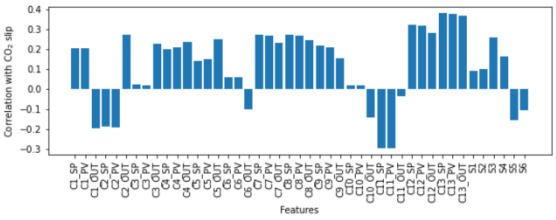

Figure 6: Correlations between the target and features

The correlations shown in Figure 6 help the selection between redundant features. SP is kept for each controller except C2 and C4, and PV is removed from the dataset based on correlations. Regarding the C2 and C4, PVs are kept instead of SPs as they have higher correlations with the target. Therefore, the number of features is reduced to 32.

3.4. Feature selection: wrapper methods

In contrast to filter methods, wrapper methods depend on the type of final model in evaluating the performance of features. Therefore, let us assume the model to be OLS or FIR. Although this assumption may lead to removing features with valuable non-linear information about the target, these models make a fair basis for feature selection for linear modeling in general.

Two common general ideas of wrapper methods are backward and forward feature selection. In addition to these standard methods, two innovative algorithms are also used in this paper. Let us explain the methods in brief:

Forward selection: several models, each including one of the features, are made. Then the feature corresponding to the best-performed model is selected as the most promising one. In the next rounds, the same procedure is used where all models include the selected variable(s) from the previous round. Finally, the algorithm stops when adding new features does not increase the model performance.

Backward selection: the algorithm starts with a model that includes all features. Then in each step, one variable whose removal leads to the highest performance of the model is selected and removed from the dataset. This procedure continues until removing none of the variables leads to a better model.

Backward-forward ver. 1: in each round, one complete forward selection procedure follows by one complete backward selection. If the forward or backward selection in one of the rounds does not change the selected set, the algorithm stops.

Backward-forward ver. 2: after each round of adding a variable to the list, one backward round is run. Therefore, each backward or forward round removes or adds only one feature.

Note that both backward-forward algorithms (i.e., ver. 1 and 2) can start from an empty or complete set. Therefore, each algorithm has two variants (i.e., empty set start and full set start).

A metric is needed for the model's performance to decide on adding or removing features in all the above-mentioned algorithms. The most common metric is the mean square error (MSE) between predicted and actual values of the target in the test set.

$$\text{MSE} = \frac{1}{N}\sum_{k=1}^{N}\left(y(k) - \hat{y}(k)\right)^2 \tag{10}$$

where $\hat{y}(k)$ is the model prediction for kth sample in the train/test set.

Another helpful metric is the R^2-Score which is the proportion of the variation in the target variable that is predictable from the features:

$$R^2\text{-Score} = 1 - \frac{\text{MSE}}{\frac{1}{N}\sum_{k=1}^{N}(y(k) - \bar{y})^2} \tag{11}$$

where $\bar{y}$ is the total mean of the target.

Using each algorithm (i.e., 6 in total) with the two mentioned metrics (i.e., MSE and R^2-Score) on each train-test pair yields 12 different sets for the features. For instance, using the forward selection algorithm and MSE metric on train-test set 1 gives the following features as the promising ones: C13_SP, C9_OUT, C6_SP, C10_OUT, C12_SP, S3. To save space, the full results are not shown here. Instead, the appearance counts of the features in the final models separated by the train-test sets are shown in Figure 7.

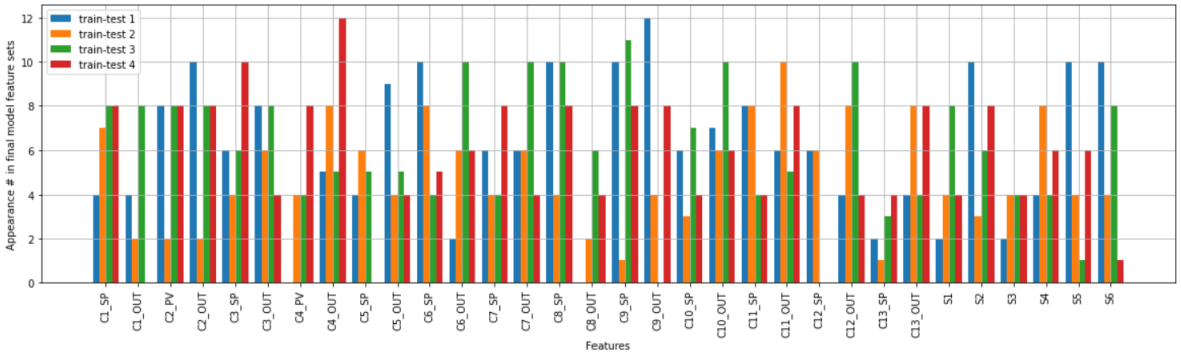

Figure 7: Appearance count of features

One first observation is that the occurrence of some features does not really show their importance. For example, feature S2 (i.e., ambient temperature) is in the final models 10, 3, 6, and 8 times. Obviously, the ambient temperature has almost no meaningful effect on the CO_2 slip values. Therefore, this suggest that most other variables do not contain more information about CO_2 slip than the ambient temperature.

Table 4 lists all variables present in more than half of the final models on average. Based on the information given in Table 4, the most important feature is C8_SP (i.e., set-point value for semi-lean solution controller), which is reasonable. The other important features are C4_OUT (i.e., primary reformer temp.), C9_SP (i.e., purge gas flow into the unit), C10_OUT (i.e., purge gas into the primary reformer), C11_OUT (i.e., syngas compressor suction pressure), and C2_OUT (i.e., steam-to-carbon ratio), where all relate to the plant load and affect the average of CO_2 slip rather than its trends. The rest of the features, as mentioned, are not more informative than ambient temperature.

Another observation from Table 4 is the difference between train-test set #2 and the other sets. This suggests a different operation regime in the plant between Sep. 6 and Sep. 12.

Table 4: The most important features based on wrapper methods

	Number of occurrences				
Feature	**Train-test 1**	**Train-test 2**	**Train-test 3**	**Train-test 4**	**Average**
C1_SP	4	7	8	8	6.75
C2_PV	8	2	8	8	6.50
C2_OUT	10	2	8	8	7.00
C3_SP	6	4	6	10	6.50
C3_OUT	8	6	8	4	6.50
C4_OUT	5	8	5	12	7.50
C6_SP	10	8	4	5	6.75
C7_OUT	6	6	10	4	6.50
C8_SP	10	4	10	8	8.00
C9_SP	10	1	11	8	7.50
C10_OUT	7	6	10	6	7.25
C11_OUT	6	10	5	8	7.25
C12_OUT	4	8	10	4	6.50
S2	10	3	6	8	6.75

4. Discussion

Based on the features found in previous section, let us make our final models for each train-test set. The models are made based on the train sets, and prediction results on the test set are shown in Figure 8, together with the measured values. The initial OSL model predictions (section 3.2) are also plotted for better comparison.

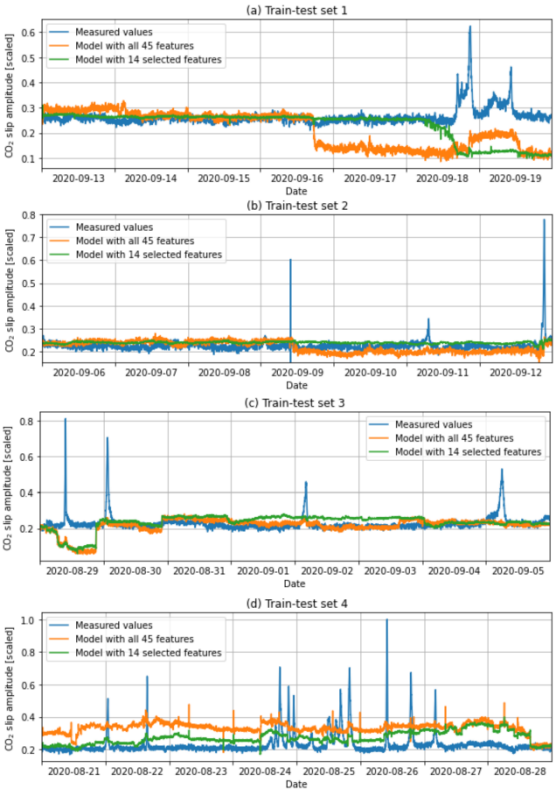

Figure 8: Performance of different models on test sets

The plot's first conclusion is that the feature selection procedure leads to models with better predictions. The second conclusion is that the models suffer from noise that is not white. In other words, some other crucial variables are not

available in the given dataset. This makes linear modeling almost impossible.

Additional attempts were made on this dataset to evaluate the correctness of above-mentioned conclusions. For example, some pre-processing methods used before applying the feature selection methods. Among them:

- the data was smoothed using moving average filter,
- Hampel filter used for detecting and removing outliers, and
- non-scaled and standardized data (i.e., scale to data to have mean of 1 and standard deviation of 0) were used instead of normalized data.

None of the pre-processing methods changed the results noticeably.

In addition, instead of Eq. (3), partial least square (PLS) method was used for finding coefficients in both static OSL and FIR models. PLS automatically reduces the number of features using SVD decomposition. However, those features would be transformed features that have no physical interpretation. Also, using PLS did not contribute to the performance of the models.

To find out how non-linear models perform on this dataset, deep learning methods such as long short-term memory (LSTM) and convolutional neural networks (CNNs) models were also tried. However, all of the models failed to follow the trends in the data. Therefore, "missing feature(s)" is the best description for the case study dataset.

5. Conclusions

The outset of this work was to develop a data-driven model for use in model predictive control (MPC). When considering such models, it is vital to know whether the data contain sufficient information for such a model. In other words: whether the data is informative. In this paper, a walkthrough for feature selection given a historical dataset measured from a process was reviewed. In addition, the informativity of the data is also checked during the feature selection process. The step-by-step method was applied to a case study dataset from an ammonia plant. However, the fact that irrelevant features such as ambient temperature are among the selected features suggests the dataset is not informative enough to predict the target variable. In addition, the prediction results show some trends that the models cannot follow. Further works (i.e., non-linear modeling approaches – not presented in the current paper) illustrate the correctness of the presented walkthrough and non-informativity of the data for modeling the process.

References

[1] P. Fritzson, Introduction to modeling and simulation of technical and physical systems with Modelica: John Wiley & Sons, 2011.

[2] O. Nelles, Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes: Springer Nature, 2020.

[3] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, Dynamic mode decomposition: data-driven modeling of complex systems: SIAM, 2016.

[4] K. H. Esbensen, D. Guyot, F. Westad, and L. P. Houmoller, Multivariate data analysis: in practice: an introduction to multivariate data analysis and experimental design: Multivariate Data Analysis, 2002.

[5] Yara Fertilizer Industry Handbook, Yara International ASA, Oslo, 2018.

[6] V. Pattabathula, and J. Richardson, "Introduction to ammonia production," CEP magazine, vol. 2, pp. 69-75, 2016.

[7] J. Gao, Y. Wang, Y. Ping, D. Hu, G. Xu, F. Gu, and F. J. R. a. Su, "A thermodynamic analysis of methanation reactions of carbon oxides for the production of synthetic natural gas," vol. 2, no. 6, pp. 2358-2368, 2012.

[8] V. Stojanovic, N. Nedic, D. Prsic, and L. Dubonjic, "Optimal experiment design for identification of ARX models with constrained output in non-Gaussian noise," Applied Mathematical Modelling, vol. 40, no. 13-14, pp. 6676-6689, 2016.

[9] I. Guyon, and A. Elisseeff, "An introduction to variable and feature selection," Journal of machine learning research, vol. 3, no. Mar, pp. 1157-1182, 2003.