Surrogate model development for hydrogen separation via pressure swing adsorption processes: selection and evaluation of machine learning algorithms

Dominik Freund[1[0000-0002-7459-2681]] and Burak Atakan[1[0000-0002-1361-8315]]

[1] University of Duisburg-Essen, Institute for Combustion and Gas Dynamics (IVG), Chair of Thermodynamics, Lotharstraße 1, D-47057 Duisburg, Germany
dominik.freund@uni-due.de

Abstract. Within complex chemical engineering applications, subsystems or technologies have to be selected and evaluated, without being an expert in each technology. Here, a surrogate model can help, if it can be set up with easily available reliable tools and data from publications. A surrogate model for pressure swing adsorption processes to separate hydrogen from gaseous mixtures was developed here, using 90 published data sets for training and testing five different machine learning algorithms. The resulting specific surrogate model is valuable, as also the procedure of its development and analysis, which is transferable to other scientific questions.
For these data sets a random forest regression yielded the best results, in terms of high coefficient of variation, low mean absolute error and low root mean square error, when 80 % of the data was used for training and 20 % for testing. The predicted hydrogen recovery deviated from the true value by 6.6 %.
A subsequent global sensitivity analysis revealed that the hydrogen recovery is mainly dependent on the number of adsorption beds and adsorption time. The purity depends on the adsorption pressure and the purge to feed ratio but should be investigated further by increasing the number of data sets, as soon as more publications become available. In the future, the surrogate model shall be implemented in a subordinate process concept model for testing the suitability of PSA processes for the separation of hydrogen from exhaust gas mixtures ordinating from fuel-rich operated HCCI engines.

Keywords: Machine learning, pressure swing adsorption, hydrogen separation, surrogate model.

1 Introduction

Often in chemical and energy engineering processes it turns out that certain sub-systems, which are considered, are not as favorable as wanted, e.g. a certain separation process. In such cases it would be very helpful to investigate alternatives, but without being an expert in the alternative procedure, the selection of good process parameters is either difficult or very time consuming. To evaluate its suitability, either costly experiments or extensive simulations could be performed. Both approaches demand either expertise in this research field or lots of effort in terms of time and workload. If the final outcome was interesting for the scientific community, this effort would have been worthwhile. However, if the outcome was that the PSA process is not feasible for the

desired application, a shortcut would have been preferred to speed up the evaluation of feasibility. Furthermore, with increasing model complexity and number of investigated parameters computational time can be challenging and surrogate models are discussed as promising solutions [1], and summarizing investigations from the literature would be helpful, if such a model can easily be set up with recent machine learning algorithms, as will be described here. But this also leads to the question, which of the different machine learning algorithms will lead to good predictions, while being robust with respect to their usage. This will be investigated here for a specific problem, assuming that not only the specific results are helpful, but also the procedure for scientists facing similar problems.

In a recent study of fuel rich HCCI processes for polygeneration that can be used to produce work, heat, and synthesis gas or hydrogen, as shown in our recent work [2–4], it turned out that the implemented hydrogen separation with a membrane is the costliest part. Thus, the question arises whether alternatives would lead to a techno-economically more favorable process with the needed purities. One choice to evaluate is pressure swing adsorption, which adsorbs CO_2, H_2O, and other impurities and provides a purified hydrogen flow [5]. However, due its dynamic behavior and large number of variables, e.g. adsorption pressure, adsorption time, purge to feed ratio, adsorption bed geometry etc., as well as its complex interaction of time- and position-dependent kinetics and thermodynamics, high efforts are needed to implement such a model in a superordinate model. Therefore, the performance of different easily available and established machine learning algorithms is evaluated with respect to the suitability of the derived surrogate model.

Although this being a field with increasing interest, there is a small number of publications about surrogate models of pressure swing adsorption processes. The first publications stem from 2017, with increasing numbers in the following years. Most authors chose single hidden layer artificial neural network (ANN) methods for multi-objective optimizing product purity and yield of the PSA process.

One of the early works was conducted in 2017 by Sant Anna et al. [6]. They trained a feedforward ANN model for N_2/CH_4 separation in a silicalite adsorbent and fed the neural network with data simulated with a vapor pressure swing adsorption (VPSA) model. Eventually, they used the ANN model for optimizing N_2 purity and recovery. Ye et al. [7] and Xiao et al. [8] used a similar approach in 2019, but with different gas mixtures and thus different adsorbents. Ye et al. investigated the hydrogen separation from a $H_2/CO_2/CO$ mixture with copper containing adsorbents (Cu-BTC). Xiao et al. used active carbon and zeolite 5A separately, and a layered bed, to separate hydrogen from a $H_2/CO_2/CH_4/CO/N_2$ mixture. Two years later, Tong et al. [9] used the same approach for H_2/CH_4 mixtures and zeolite 5A as adsorbent. They also investigated the influence of the number of neurons in the hidden layer and the number of samples on the surrogate model performance.

Neural network surrogate models for PSA were also used for analyzing CO_2 capture by Leperi et al. [10] and Subraveti et al. [11] in 2019. Leperi et al. trained the neural network with data from partial differential algebraic equation model simulations, using Ni-MOF-74 and zeolite 1X as adsorbents. They found that the highest relative error for CO_2 purity and recovery was 1.42 % and thus concluded that the PSA process model can be accurately substituted by a well-trained neural network.

In 2022 Rebello et al. [12] summarized some of the above mentioned publications that used neural networks and concluded that those methods are often applied without

evaluation of their limitations and the challenges the predictors entail. Rebello et al. therefore performed PSA experiments for CO_2 capture from $CO_2/CO/H_2$ mixtures and trained three different neural network models with the data: feedforward neural network (FNN), recurrent neural network (RNN), and deep neural network (DNN). Their objective was to accurately predict the time dependent evolution of CO_2 purity and CO_2 recovery and to provide guidelines to creating neural network based surrogate models. Therefore, they firstly optimized the hyperparameters of the neural networks, which are the parameters that define the structure of the neural network. Afterwards, they trained the models and evaluated the coefficient of determination (R^2), the mean absolute error (MAE), and the root mean square error (RMSE) of the predictions. The FNN model showed the best results for predicting the measured data, whereas the DNN model predicted the simulation results better. They concluded that for choosing the most suitable machine learning model it is crucial to consider the application of the model.

There are only few works on surrogate models based on other methods than neural networks. One example is the work of Pai et al. [13] from 2020. They trained and compared five different machine learning methods to model a 4-step VPSA for CO_2 separation from CO_2/N_2 mixtures on 13X zeolite. The methods were decision tree regression (DTR), random forest regression (RFR), support vector regression (SVR), gaussian process regression (GPR), and artificial neural networks (ANN). The data for model training was obtained with a detailed VPSA model and 800 data sets were created. Eventually, they used the surrogate models to optimize the CO_2 purity and recovery, as well as the energy consumption and productivity. They found that the GPR model predicted their simulated data most accurately and matched their experimentally determined purity and recovery within an error of 5 %.

Hao et al. [14] developed a methodology to efficiently generate a surrogate model for any existing model by using a support vector machine (SVM) classifier to reduce the amount of simulated data for surrogate learning to a reasonable minimum. They could thus reduce the time to generate surrogate models by 86 % for a PSA model and 51 % for a gas-to-liquids model.

Yan et al. [15] investigated the separation of O_2 from N_2/O_2 mixtures via PSA to find the best suitable adsorbent material of different metal-organic frameworks. They trained three different machine learning models, namely random forest, gradient boost regression tree, and extreme gradient boosting, with 6,013 different metal-organic framework variants. With extreme gradient boosting they achieved the highest coefficient of determination ($R^2 = 0.93$) and the lowest mean absolute and square mean errors.

The aforementioned authors typically are experts on pressure swing adsorption processes and their objective was mostly to save computational time of processes they already modelled in detail or investigated experimentally.

In this work, the approach is different. The objective is to model the hydrogen purity and hydrogen recovery of pressure swing adsorption processes as a function of important input parameters and to implement this correlation as a surrogate model in our polygeneration process model, as a mean to separate hydrogen from the exhaust gas containing a mixture of $CO_2/CO/H_2/H_2O/N_2$. Therefore, machine learning methods are helpful as well: data can be extracted from the literature and used for training and testing of the model. The performance is expected to be poorer due to the greater divergence between data sets of different works. Nevertheless, in this work the question shall be answered if machine learning based surrogate models are feasible for evaluating the suitability of PSA processes with acceptable accuracy and limited data from the

literature. Furthermore, it is discussed which approach and machine learning method seems to be reasonable. Eventually, the model with the best performance was chosen to perform a global sensitivity analysis to identify the sensitivity of the input values on the output values.

2 Methods

In pressure swing adsorption a gas mixture flows through a tube filled with adsorbent of a certain length at a given pressure and temperature. Due to differences in their adsorption behavior, the outlet partial pressures change with time and differ from their inlet values; this can be used for separation. The purity of the exiting gases depends on different parameters like adsorbent, time, flow velocity, pressure, tube length and diameter, initial composition, and temperature. After a certain period, the adsorbent is saturated and the outlet partial pressures are the same as the inlet partial pressures, without a separation. Before this happens, generally a second adsorbent tube is used for separation. The adsorbent can be refreshed by reducing the pressure and purging with an inert gas. Generally, several adsorbent tubes are used to ensure a steady state operation and often combinations of two to three different adsorbents in a layered bed are applied to achieve a certain separation goal, but also up to six adsorbents are discussed [16]. This leads to the variables which must be modeled either as input or as output to the machine learning algorithm.
The machine learning algorithms in this work were trained and tested in a python framework using the *scikit-learn* module *sklearn* [17]. In this section, the workflow of the python program is presented, the investigated machine learning methods are briefly explained and compared, and the defined input and output variables of the machine learning model are given. **Fig. 1** shows the workflow which illustrates the approach in this work.

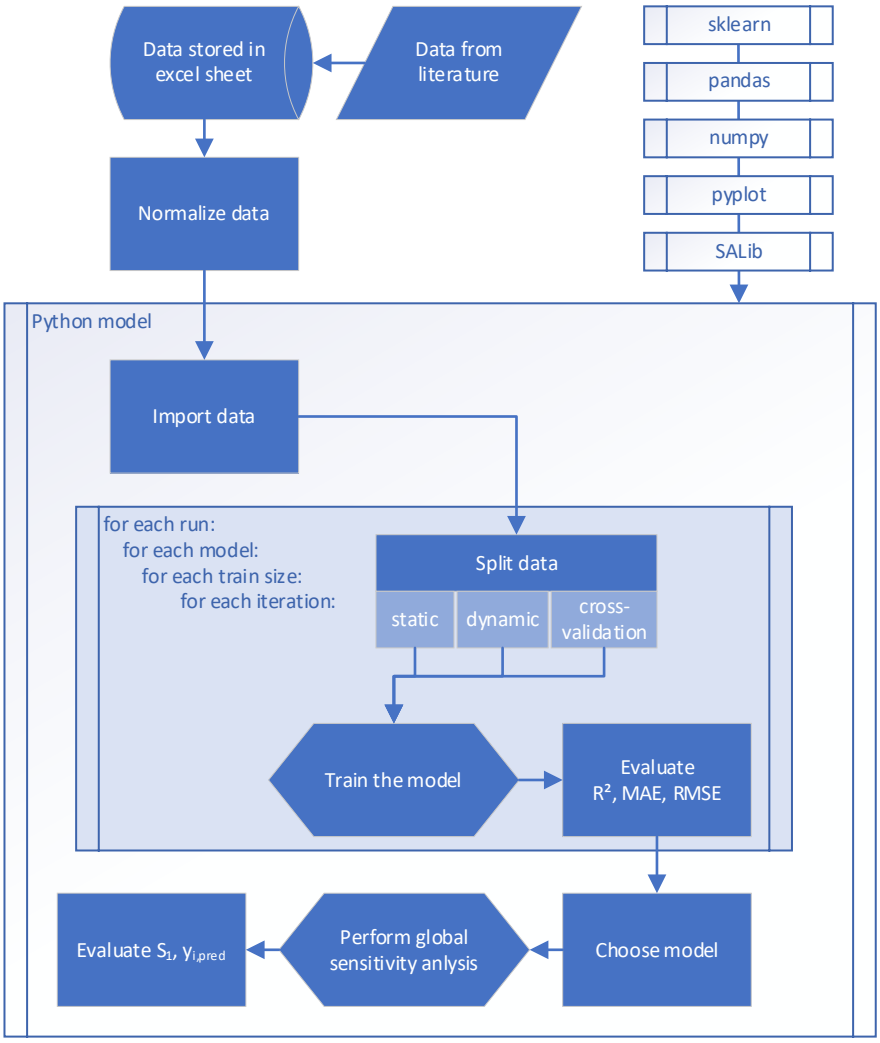

Fig. 1. Schematic of the workflow.

The first step is to compose a structured table containing the data from the literature according to some criteria: the adsorbents should be active carbon and/or zeolite and the separation product must be hydrogen, separated from a gaseous mixture. Additionally, the following parameters should be given:

- adsorption pressure (p_{ads})
- desorption pressure (p_{des})
- hydrogen recovery (R_{H2})
- hydrogen purity (P_{H2})
- feed temperature (T_F)
- number of adsorption columns (N_c)
- mole fraction of hydrogen in the feed (x_{H2})
- purge to feed ratio (P/F)

- active carbon to zeolite ratio (AC/Z)
- adsorption time (t_{ads})

Eventually, 90 suitable data sets from the literature were found [8,18–23], 52 of them were determined experimentally and 38 of them were simulated. The data can be found in Table A1. These data sets were used to build a surrogate model. The objective of this model was to find a correlation between the input variables x_i and the output variables y_i, that predicts the behavior of the experiments or detailed models with an acceptable accuracy. Therefore, the input variables and output variables have to be defined. The input variables can also be called predictors, features, or independent variables, whereas synonyms for the output variables are target variables, responses, or dependent variables [24] p. 15. Most of the input variables have different units and thus different orders of magnitude, e.g. adsorption time is given in *s* and adsorption pressure in *bar*. As a consequence, some algorithms require the data of the input variables to be normalized, which was done according to eq. (1) [11].

$$\bar{x} = \frac{x - min(x)}{max(x) - min\,(x)} \tag{1}$$

The normalized data was subsequently imported to python using the module *pandas* [25]. The imported data was then split either into a training and a test sample, defined by the train size which gives the amount of data sets used for training, or by using a five-fold cross validation method. Schenker and Agarwal [26] pointed out that splitting is a crucial step for training machine learning algorithms, especially when only small data sets are available. They also showed that cross-validation for splitting the data sets can increase the accuracy of artificial neural network models. The idea of cross-validation is that the data sets are randomly split into train and test set multiple times to assure that each data set was once used for testing. A similar approach is described by Kramer [27] p.18.: a train, test, and validation set are defined. The purpose of the validation set is to be left out by the cross-validation and eventually test the resulting model.

In this work, we evaluated the machine learning models by training them 20 times and by performing a five-fold cross-validation, implemented in *sklearn*.

The models were evaluated by calculating the coefficient of determination R^2 [28], the root mean square error $RMSE$ [29,30], and the mean absolute error MAE [29] of the predicted target values, according to eq. (2), (3), and (4).

$$R^2 = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2} \tag{2}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{3}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{4}$$

The coefficient of determination R^2 compares the variance of the predicted output variable $\hat{y}_i$ with the variance of the true output value y_i [28], hence a value of $R^2 = 1$ expresses that the scatter of the predicted data is exactly the same as the scatter of the true data. According to Jiang [31] p. 98, R^2 values below 0.5 signify a weak correlation between predicted and true output. An R^2 value between 0.5 and 0.8 indicates an inaccurate model, possibly due to large measurement spreads of the true data used for training. Jiang also points out that a R^2 value higher than 0.8 must not necessarily mean that the model is more accurate, since if irrelevant data was used for training as well, R^2 might increase without increasing the models predictive performance. R^2 can therefore not be the only metric to evaluate machine learning models.
The second metric is the root mean square error, which describes the quadratic mean deviation of the predicted output $\hat{y}_i$ from the true output y_i. The mean absolute error is a similar measure but here, for the deviation of $\hat{y}_i$ and y_i, the absolute value is calculated. In 2005, Willmot et al. [29] strongly advocated the use of MAE instead of RMSE , since they found that the RMSE values are ambiguous, have thus no clear interpretation, and are typically higher than the MAE values. However, since RMSE is still an often-used metric, we also used it here – together with R^2 and MAE – to obtain comparability with other works.

2.1 Description of the investigated machine learning algorithms

Machine learning methods can be distinguished between parametric and non-parametric methods [24] pp. 21. A parametric model assumes a function relating output to input variables and subsequentially fits the function and trains its parameters. The advantages are a simplified problem and easy, time-saving fitting. However, this is only applicable if the form of the function is known. Here, for pressure swing adsorption processes, this is not the case and thus all methods utilized in this study are non-parametric methods. It must be mentioned that the disadvantage of non-parametric methods is the requirement of more data points compared to parametric-methods and the tendency to overfitting. In this section, the investigated algorithms are shown in **Table 1** and subsequently, they are briefly described.

Table 1. Investigated machine learning algorithms and their most important parameters (default values of *sklearn* are shown and also used in this work, if not stated otherwise).

Abbr.	Algorithm name	Hyperparameters (excerpt)		References
SVR	Epsilon-Support Vector Regression	epsilon	0.1	[32]
		C	1.0	
		kernel	‘rbf’	
		γ	‘scale’	
		r	0	
		d	3	
DTR	Decision Tree Regressor	max_depth	∞	[33]
		min_samples_split	2	
		min_samples_leaf	1	
		criterion	‘squared’	
RFR	Random Forest Regressor	max_depth	∞	[34]
		min_samples_split	2	

		min_samples_leaf	1	
		criterion	'squared'	
		n_estimators	100	
MLPR	Multi-layer Perceptron Regressor	hidden_layer_sizes	100	[35]
		activation	'logistic'	
		learning_rate	0.001	
		max_iter	1500	
KNR	K-Nearest Neighbors Regression	k	5	[36]
		weights	'uniform'	
		algorithm	'auto'	

There are many non-parametric methods, e.g. MARS[1], PRIM[2], and variations of the investigated methods, described comprehensively by Hastie et al. [30]. To reduce the number of models, only methods that are often used in the literature were chosen for this study. Linear regression and gaussian process regression, which are parametric methods, were also tested but since they led to near-zero and negative coefficients of determination, they are not discussed further here.
Changing the hyperparameters changes the structure of the model. Therefore, for improving the model accuracy, optimization of these parameters is crucial. However, optimization is not within the scope of this work and random changes to the most important hyperparameters showed no significant improvements. Consequently, the default parameters implemented in *sklearn* are assumed to be reasonable enough for checking the performance of the different algorithms.

Epsilon-Support Vector Regression (SVR). The main idea of ε-support vector machines is that a regression is performed with an acceptable error margin of ±ε that the predicted target values may deviate from the targets [37]. Additionally, the hyperparameter C defines the tolerance of target values lying outside of the error margins, see Christianini and Shaw-Taylor [38] (pp. 114). Therefore, with a high C the algorithm aims for predicting a large amount of the targets, whereas a small C denotes for a small tolerance of exceeding or falling below the boundaries of the target value plus/minus ε.

In support vector machines, kernel functions are used to perform a non-linear calculation without increasing the number of hyperparameters [38] (pp. 26). In sklearn, the following kernels can be chosen: linear, polynomial ('poly'), gaussian radial basis function ('rbf'), sigmoid, or precomputed and should be chosen according to the data. Here, the (default) radial basis function is used because it yielded the highest R^2 values at a random test with a train size of 0.5. Equations (5) to (8) show the kernel functions included in *sklearn* [39].

$$linear: \langle x, x' \rangle \tag{5}$$

$$polynomial: (\gamma \langle x, x' \rangle + r)^d \tag{6}$$

[1] Multivariate Adaptive Regression Splines
[2] Patient Rule Induction Method

$$\textit{radial basis function:}\ \exp\left(-\gamma\left|\left|x-x'\right|\right|^2\right) \tag{7}$$

$$\textit{sigmoid:}\ \tanh(\gamma\langle x,x'\rangle + r) \tag{8}$$

The hyperparameter γ defines the influence of a single data set on the training. With high γ single data sets effect the training strongly and vice versa. The parameters r and d denote for a scalar coefficient and the degree of the polynomial, respectively. A data normalization is required in advance and the SVR is only single-output capable. Therefore, the *MultiOutputRegressor* class from *sklearn* is used [40], which fits one regression per output value and thus makes single-output regression methods multi-output capable.

Decision Tree and Random Forest Regressor (DTR/RFR). A Decision Tree is a machine learning algorithm which consists of nodes and branches. The input variables are split according to specific criteria until all leaf nodes (which are the last ones of a branch) are "pure" [41]. A pure leaf node means that it yields exactly the target value. In the *sklearn* algorithm, the input variables are split until the maximum depth is reached, all leaves are pure, or all leaves contain a defined number of samples [33].
Some of the advantages of decision trees are their capability of handling data with missing values [41], they are multi-output capable, and no data preparation, e.g. normalization, is required. On the contrary, they tend to overfitting if the settings (minimum number of samples, maximum depth of the tree) are not tuned accordingly. It must also be mentioned that they are poor at extrapolations.

To overcome the issue of overfitting and increasing the accuracy, random forest methods have been developed. These are so called "ensemble methods" that use multiple algorithms to improve the prediction performance. Here, the random forest regressor utilizes multiple decision tree regressions and averages the outcomes [30] pp. 587.

Multi-layer Perceptron Regressor (MLPR/NN). A multi-layer perceptron is a specific type of artificial neural networks. A neural network is two step regression or classification model [30] p. 392. It consists of an input layer containing the input variables, a single or multiple hidden layers, and the output layers with the model responses. Each layer contains nodes which transform the data they receive. The nodes in the hidden layers (between input layer and output layer) must be activated to pass a transformed value. Several activation functions can be chosen in *sklearn*: 'identity', 'logistic', 'tanh' and 'relu'. Typically, a sigmoid or S-shape function, e.g. f(x) = 1 / (1 + exp(-x)) is utilized according to Hastie et al. [30] p. 392. This corresponds to the 'logistic' function in *sklearn* and is thus chosen for this work.

To fit the neural network, unknown parameters, called weights, must be calculated. This can be achieved by minimizing the sum-of-square errors R between target value y_{ik} and model response $f_k(x_i)$ according to Eq. (9) [30] p. 395.

$$R(\theta) = \sum_{k=1}^{K}\sum_{i=1}^{N}\left(y_{ik} - f_k(x_i)\right)^2 \tag{9}$$

Since this would probably lead to overfitting, the back-propagation method is often used and also used in this work by choosing the solver ‘adam’ in *sklearn*. The idea of back propagation is to compare the output values of the neural network with the correct value and adjusting the weights iteratively, so that the error function (9) is step-wise reduced until the calculation converges. To achieve this, the gradient descent method is applied which includes, additionally to eq. (9), the derivatives of eq. (9) by the weights. These derivatives contain the learning rate γ, a hyperparameter for tuning the training of the neural network.

Regression based on k-nearest neighbors (KNR). James et al. state that k-nearest neighbor regression is one of the simplest non-parametric methods and often one of the most accurate methods [24] p. 104. The approach is to average the k-nearest neighbors in the training data set, with k being a user-defined integer hyperparameter called “neighborhood size”. Values of k can be positive integers of 1 to N, whereas N denotes for the number of data points. For small k the model tends to overfitting, whereas for high k the averaging is smoothed [27] pp. 15.

2.2 Brief comparison of the utilized algorithms

Table 1 compares the algorithms for some characteristics, as discussed in [30] by Hastie et al. Additionally, the authors emphasize that even with that characterization the suitability of the algorithms for the specific task cannot be known in advance.

Table 2. Machine learning algorithm characteristics compared. The data is taken from [30] p. 351. Key: ▲ = good, ◆ = fair, ▼ = poor.

Characteristic	MLPR/ NN	SVR	DTR/ RFR	KNR
Natural handling of data of ‘mixed’ type	▼	▼	▲	▼
Handling of missing values	▼	▼	▲	▲
Robustness to outliers in input space	▼	▼	▲	▲
Insensitive to monotone transformations of inputs	▼	▼	▼	▼
Computational scalability	▼	▼	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	◆
Interpretability	▼	▼	◆	▼
Predictive power	▲	▲	▼	▲

The robustness to outliers in input space is assumed to be of high importance due to the different origins of input data, which increases the probability of outliers. Therefore, decision tree, random forest, and k-near neighbor regressions perform well according to Hasti et al, whereas neural networks and support vector regressions have poor characteristics. Furthermore, the ability to deal with irrelevant inputs may be important,

because the interdependencies of the variety of PSA parameters is unknown in advance. For this purpose, decision tree and random forest regression are the only algorithms with good performance. In contrast, the interpretability is fair or poor for all methods, which is typical for non-parametric methods, because the relationship between x_i and y_i cannot be observed [24] p. 25. Furthermore, the predictive power of all methods is good, except decision tree and random forest regressions. Therefore, the latter do not predict output values well if the predicted data lies outside the trained data sets.

2.3 Studied parameters

Before training the models, the input and output values must be defined. Here, we investigated three different cases (called runs in the following), with different input and output values. **Table 3** illustrates the input and output values chosen for the different runs.

Table 3. Studied parameters for pressure swing adsorption data. X denotes input and O denotes output values of the machine learning model.

Run	p_{ads}	R_{H2}	P_{H2}	T_F	N_c	x_{H2}	$\frac{AC}{Z}$	$\frac{P}{F}$	t_{ads}	Train size[1]	Iterations
1	x	o	o	-	x	x	x	x	x	0.1 – 0.095	20
2	x	o	x	-	x	x	x	x	x	0.1 – 0.095	20
3	x	o	x	-	x	x	x	o	o	0.1 – 0.095	20

For all runs the intake temperature is not considered as variable, since in the data sets it only varies slightly between 25 and 35 °C.

Run 1 is motivated by the choice of Xiao et al. [8], who defined adsorption pressure p_{ads}, adsorption time t_{ads}, and active carbon to zeolite ratio (AC/Z) as input values and hydrogen purity P_{H2} and hydrogen recovery R_{H2} as output values.

In this work, additional parameters must be used as well, as they vary in the different publications on which the data sets in this paper are based:

- number of adsorption columns (N_c)
- mole fraction of hydrogen in the feed (x_{H2})
- purge to feed ratio (P/F)

This leads to a total of six input values and two output values for run 1. The only difference between run 1 and 2 is that purity is an additional input value instead of an output value. The idea is that in this case it is possible to specify the required purity to be attained with the surrogate model. Moreover, there is always a trade-off between purity and recovery – which is why surrogate models for PSA process are often used for optimizing a weighted sum of purity and recovery, as discussed in the introduction. Therefore, it is assumed that the machine learning models' accuracies in this work gain from decreased complexity. Since suitable P/F ratios and adsorption times are often unknown before detailed modelling, it may also be interesting to get these values as responses of the surrogate model. Therefore, in contrast to run 2, run 3 contains those values as output values.

2.4 Sensitivity analysis

The trained models were finally tested by conducting global sensitivity analyses to investigate the influence of the input parameters on the output parameters and to assess the meaningfulness of the choice of input and output values.
A sensitivity range of ±50 % for each mean input value was chosen to consider the entire parameter space and 2^{10} samples were created for each analysis. The first order and total order sensitivity indices were calculated with the Sensitivity Analysis Library (*SALib*) [42] in Python. This method is based on the Sobol method described by Saltelli et al. in 2008 [43], which is a Monte Carlo method for generating several thousands of different, randomized values for each parameter within the chosen uncertainty or variation range. In a global sensitivity analysis, the dependence of the variance of the distribution of the output parameter is mainly decomposed with respect to the importance of the variance of the different input parameters, as explained in detail in [43]. In **Fig. 2** an example of the results of this method is shown for an HCCI engine polygeneration process concept that we investigated previously [44]. It illustrates the normalized hydrogen costs as a function of normalized operating hours. As the linear fit indicates, a sensitivity was found because the hydrogen costs depend on the operating hours.

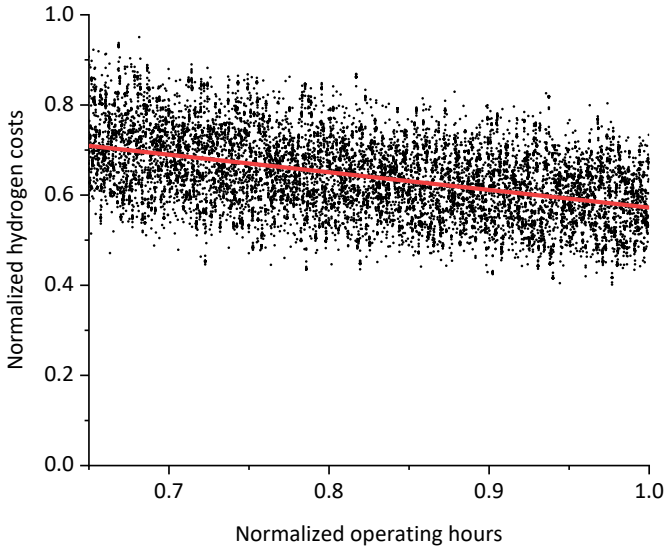

Fig. 2. Example of a scatter plot showing the normalized hydrogen costs as a function of normalized operating hours (black dots: hydrogen costs; red line: linear fit). Data is taken from our previous work on an HCCI engine polygeneration process concept [44].

Resulting scattered data is evaluated by calculating the first order sensitivity index S_i according to [43], which compares the variances of the inputs and outputs (10).

$$S_i = \frac{V_{X_i}(E_{X_{\sim i}}(Y|X_i))}{V(Y)} \tag{10}$$

V_{X_i} and $E_{X_{\sim i}}$ denote the variance and the expected value of the output value Y for a randomly changed input value X_i. The variance V_{X_i} is compared to the general variance of the output $V(Y)$. Therefore, if the resulting sensitivity index is near to one, the output is mainly dependent on this single input variable. Hence, a low sensitivity index indicates a weak dependency.

3 Comparison of machine learning algorithm performances

In this chapter, the true and the predicted outputs for a fixed train size of 50 % are compared for each method and the parameter set of run 1. Subsequently, the train size was varied, and the training was performed 20 times to find the method with the best characteristic values. A five-fold cross-validation was also performed for all methods and the three different runs (see **Table 3**). Eventually, one model was chosen for a sensitivity analysis, which is discussed in chapter 4.

3.1 General fitting for a fixed train size of 50 % (run 1)

Firstly, the data sets were split into 50 % training data and 50 % test data. Afterwards the models were trained once. The predicted values $\hat{y}_i$, hydrogen recovery and purity, are plotted as a function of the literature values y_i in **Fig. 3**.

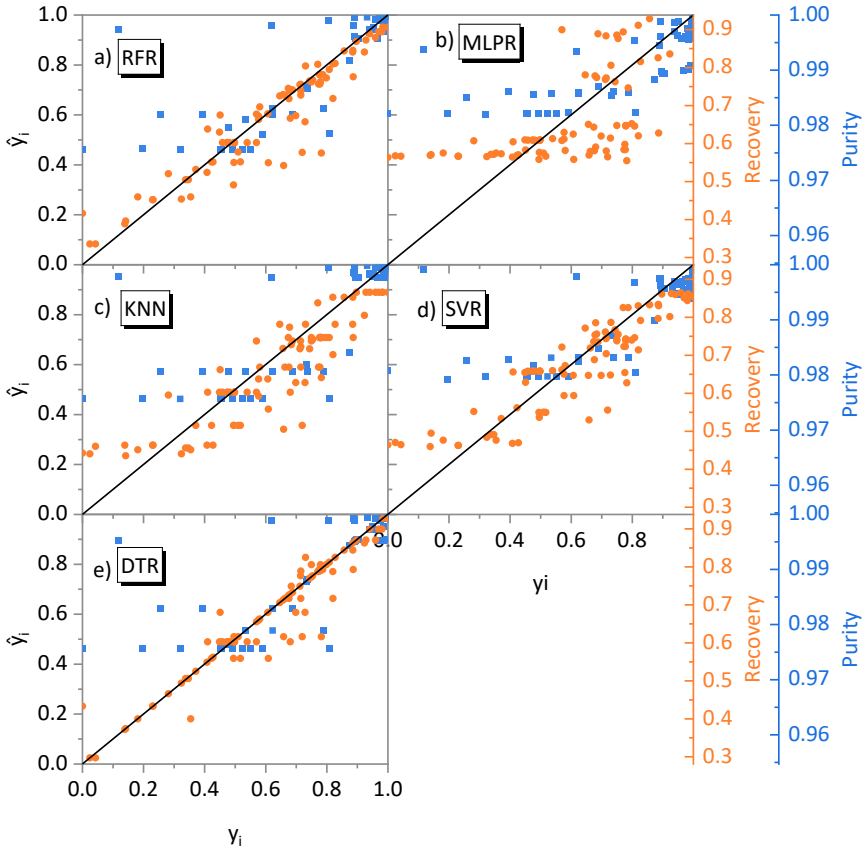

Fig. 3. True values from the literature and predicted values of the different machine learning models at a constant train size of 50 %. Orange circles represent hydrogen recovery and blue squares represent hydrogen purity.

On one hand, the recovery values (orange circles) are overall predicted much more accurately than the purity values (blue squares). On the other hand, the absolute range of the recovery is much higher compared to the purity. Furthermore, high purities of 0.99 and higher are reasonably predicted by all methods.

The lowest accuracy is achieved with the neural network (MLPR) since there are areas with similar predicted values of recovery and purity. This results in the lowest R^2 (0.45) and highest MAE (0.131) and $RMSE$ (0.174) values in this comparison. It must be emphasized that the tuning of the hyperparameters of neural networks is crucial and one of the most complex ones in this comparison. Consequently, an optimization of the hyperparameters could improve the results, but this was not in the scope of this work.

The K-nearest neighbors and SVR method perform significantly better, but the same tendencies in the deviations are obvious. The lower recovery and purity values do not follow the $\hat{y}_i = y_i$ line accurately. This results in R^2 values of 0.717 and 0.688 (MAE: 0.077, 0.093 and $RMSE$: 0.126, 0.132).

The decision tree regression performs even better. Here, the lower recovery values of 0.3 to 0.6 are accurately predicted. Since these predicted values are close to the true values, this could be a sign of overfitting. The purity accuracy is only slightly better

compared to the KNR and SVR method. The decision tree regressor yields the best accuracy in this comparison: R^2 = 0.81, MAE = 0.043 and $RMSE$ = 0.100.
Since the random forest regression is based on decision trees, the results for this method are similar. The most significant difference is that the lower recovery values deviate slightly more from the true values, resulting in a smaller R^2 of 0.798 and higher MAE and $RMSE$ of 0.052 and 0.105, respectively. However, this difference is assumed to be negligible.

3.2 Performance comparison for different splitting of train and test data (run 1)

In this section, the values for R^2, MAE, and $RMSE$ are compared for each method as it was done in the previous section, but the train sizes were varied from 0.1 to 0.95 with a step size of 0.05. Each training was performed 20 times. In **Fig. 4** the mean coefficients of determination R^2 and their deviations are illustrated, and **Fig. 5** shows the corresponding mean MAE and $RMSE$ values.

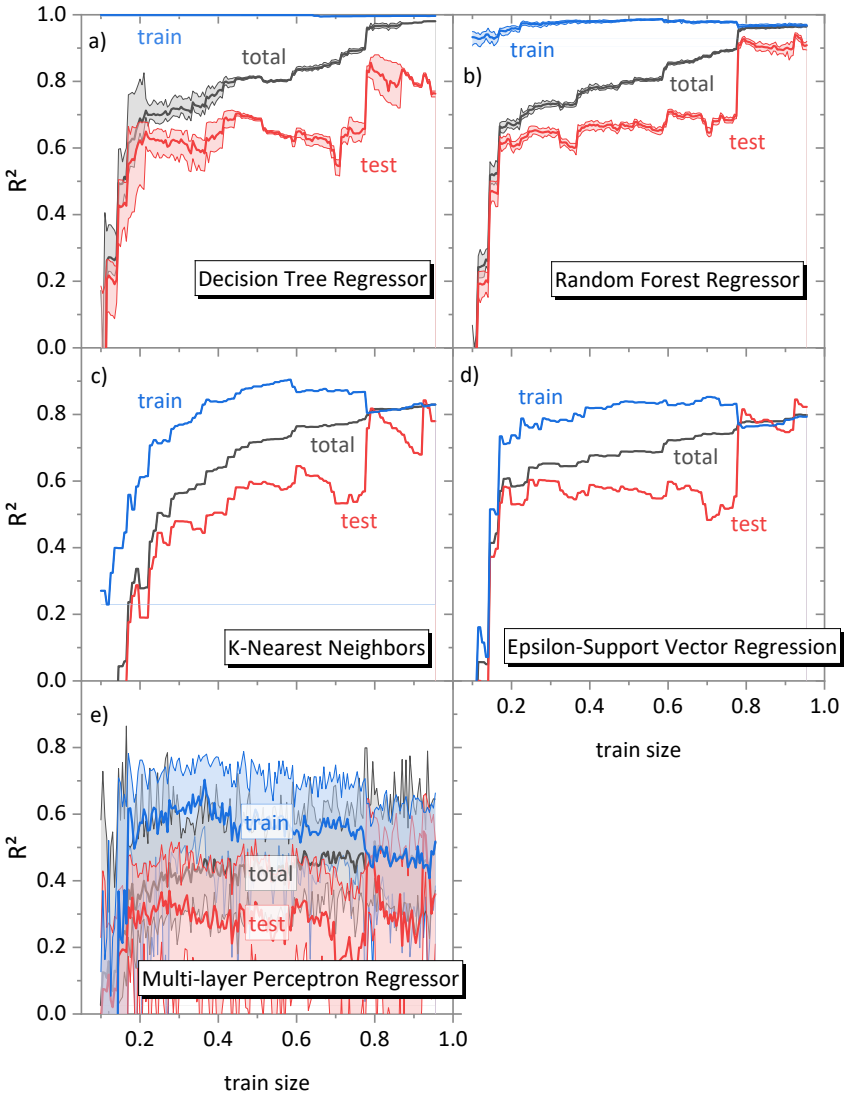

Fig. 4. Mean coefficient of determination R^2 for the five machine learning methods as a function of train size. For each train size 20 training sessions are performed, and the error bands illustrate the maximum and minimum deviations from the mean values.

Neural network, decision tree, and random forest regression show deviations from the mean values, which k-nearest neighbors and support vector regression do not. The highest deviations were found for the multi-layer perceptron regression for all train sizes. The highest mean score for MLPR is achieved at a train size of 0.925 with a poor value of 0.548; but at this point the test score is higher than the train score and the total score is still poor with a mean value of 0.454. Therefore, these results together with the huge deviations indicate that the chosen neural network is, at least without hyperparameter optimization, not suitable for training the data set in this work.

K-nearest neighbors and support vector regression perform significantly better with mean total scores of up to 0.83 and 0.798, respectively. However, the decision tree and the random forest regressor show slightly higher test scores in a huge train size area of 0.3 to 0.95 and the train scores are much better and reach values close to one.
Since the R^2 only characterize the spread of the data but gives no information about the actual deviation of the predicted values from the true values, the mean absolute errors und the root mean square errors were evaluated, illustrated in **Fig. 5**.

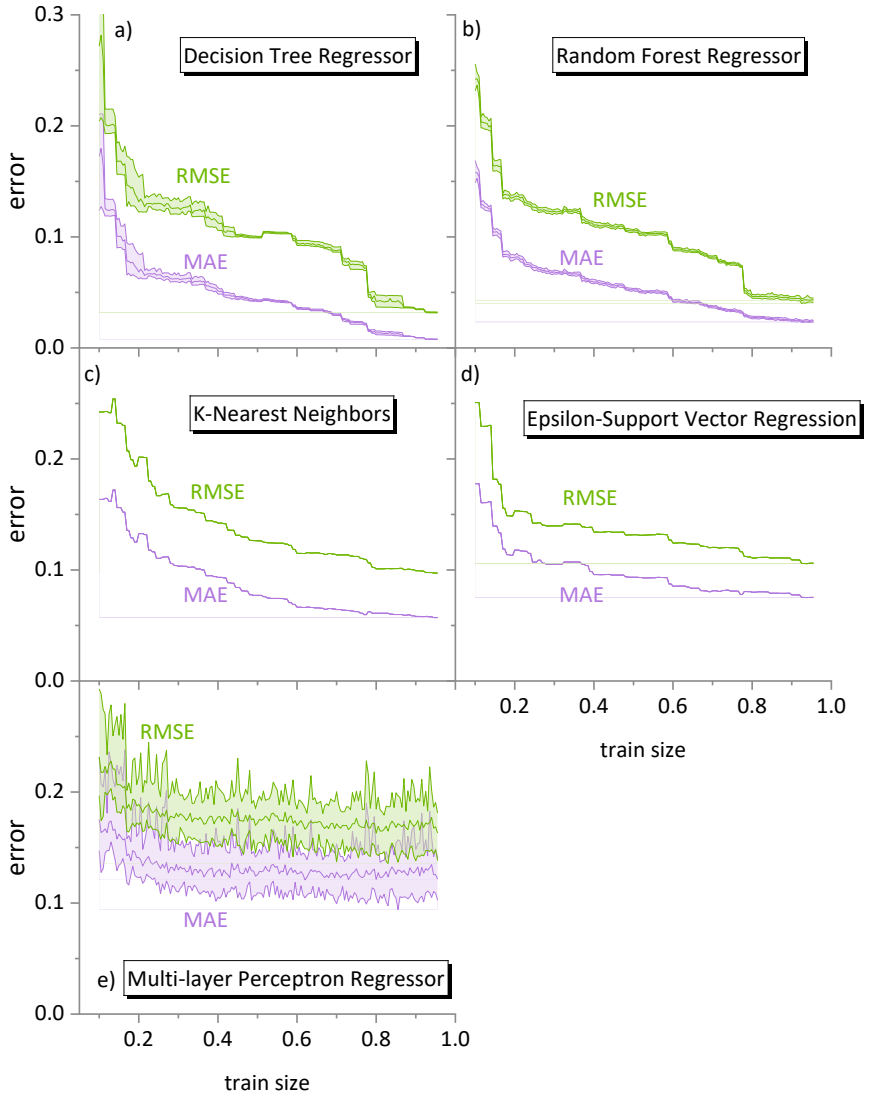

Fig. 5. Mean absolute error (MAE) and root mean square error ($RMSE$) for the five machine learning methods as a function of train size. For each train size 20 training sessions are performed, and the error bands illustrate the maximum and minimum deviations from the mean values. It must be noted that the values correspond to the normalized hydrogen recovery and purity values, which thus vary between zero and one.

As expected, the deviations of the MAE and $RMSE$ values from the mean values are comparable to the deviations discussed before for the mean R^2 values. From **Fig. 5** it can also be concluded that lower R^2 values correspond to higher errors and thus higher MAE and $RMSE$ values. The errors decrease with increasing train size for all methods, since the amount of train data increases, which achieves the highest R^2 scores and accuracy, as discussed before.

For a train size of 0.8 the random forest regression model shows high mean train, test, and total R^2 values of 0.965, 0.92 and 0.961, respectively. The mean MAE and $RMSE$ values are 0.027 and 0.046, respectively. It must be noted that the errors refer to the normalized hydrogen recovery and purity values.

3.3 Cross-validation for the three runs

The coefficient of determinations R^2 of a five-fold cross-validation for the five machine learning methods are illustrated in **Fig. 6**. In general, the train scores are higher than the test scores, which is expected. The poorest score is achieved with the neural network model for each of the three investigated runs. The best scores are achieved with decision tree regressor and random forest regressor models for all cases with a maximum value of 0.999 (run 2, decision tree regressor).

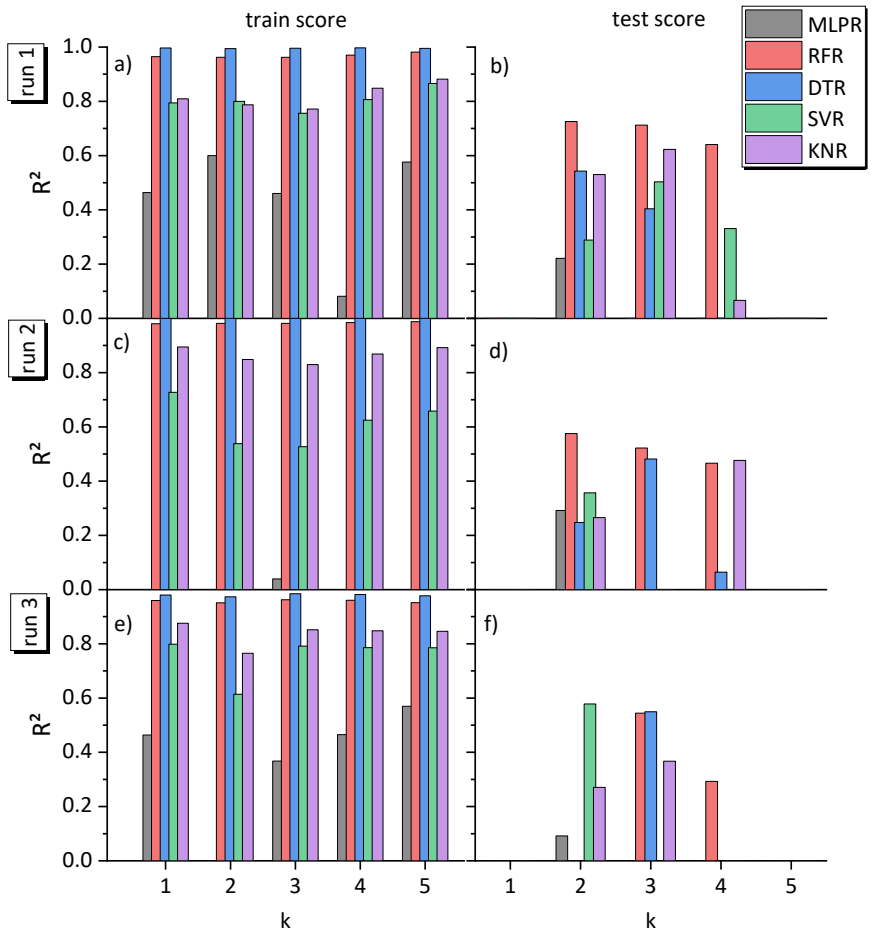

Fig. 6. Coefficient of determination R^2 as a function of cross-validation k for the training and test data. The methods are distinguished by differently colored bars. If no bar is shown, R^2 is zero or negative.

For k = 1 and k = 5 none of the models achieve positive R^2 values and thus the deviation of the predicted values from the true values is expected to be very high. For k = 2, 3 and 4, the random forest regression model performs best with R^2 values from 0.29 to 0.73; with one exception at run 3 and k = 2 where no positive values were found. The overall low scores indicate that cross-validation is problematic when only a small amount of data is available for training because the data is divided into three data sets. In the following section, the data is therefore divided into two data sets only and the split size is defined by the train size value. Since the random forest regression achieved the best results in this comparison, as seen before, it was chosen for further analysis.

3.4 Random forest regressor performance with random data splitting (all runs)

Fig. 7 illustrates the R^2, MAR, and $RMSE$ values for the random forest regression model as a function of train size for the three investigated runs. A major difference to the findings of **Fig. 4** and **Fig. 5** are the increased deviations from the mean values. These result from the random choice of the splitting into train and test data for each of the 20 iterations.

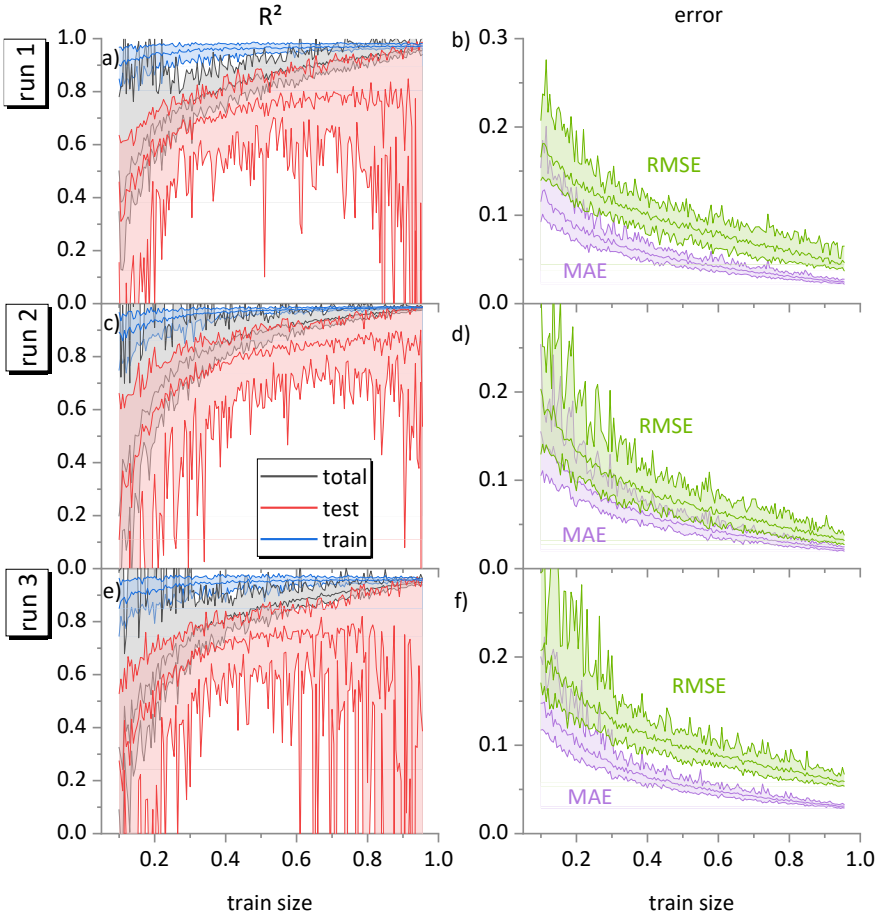

Fig. 7. Coefficient of determination R^2, mean absolute error MAE, and root mean square error ($RMSE$) as a function of train size for the three different runs and a single regression model (random forest).

In general, the R^2 values degressively increase and the MAE and $RMSE$ values degressively decrease with increasing train size – with one exception: for run 3 the mean R^2 values decrease at a train size of 0.7 and above and the deviations from the mean values increase significantly. This indicates that at very large train sizes the training data set is either too small or the target values cannot be accurately predicted with the chosen input and output values. For run 2 the best results were observed: the mean test R^2 was found between 0.85 and 0.868 at train sizes of 0.6 to 0.8. For run 1, the R^2 values are up to

0.093 or 10.9 % lower. The MAE and $RMSE$ values for run 2 vary between 0.042-0.03 and 0.063-0.046, respectively. For run 1 the root mean square errors are up to 0.017 (+37 %) higher, but the mean absolute errors are comparable. A mean absolute error of 0.03 to 0.042 is assumed to be low enough to utilize this surrogate model in a subordinate process concept. A deviation of 0.042 from a true recovery of 0.7, for instance, would mean a deviation of the predicted value by 0.064 or 9.1 % (the normalized true recovery of 0.7 is 0.741; therefore, a normalized predicted accuracy of 0.783 corresponds to a predicted recovery of 0.764). At a train size of 0.8 the MAE is 0.03 which is a deviation of 0.046 or 6.6 %.
Since the deviations for run 3 are large, this run is not considered a viable option for modelling a PSA process with the given data. From the comparison of the three runs, it can be concluded that reducing the amount of output variables increases the accuracy of the trained model, if only small number of data points are available for training.

4 Sensitivity analysis

For all three runs the first order sensitivity indexes are evaluated for the random forest regressor model discussed before, with a constant train size of 0.8. The sensitivity on four different output variables were investigated: recovery, purity, adsorption time, and purge to feed ratio, which vary between the three different runs (see **Table 3**). The total order sensitivity indexes did not significantly vary from the first order sensitivity indexes; thus, they are not discussed.

4.1 Hydrogen recovery (all runs)

According to **Fig. 8**, in each case the number of adsorption beds showed the highest sensitivity on the hydrogen recovery: 0.41, 0.32, and 0.76 for run 1, 2, and 3, respectively.

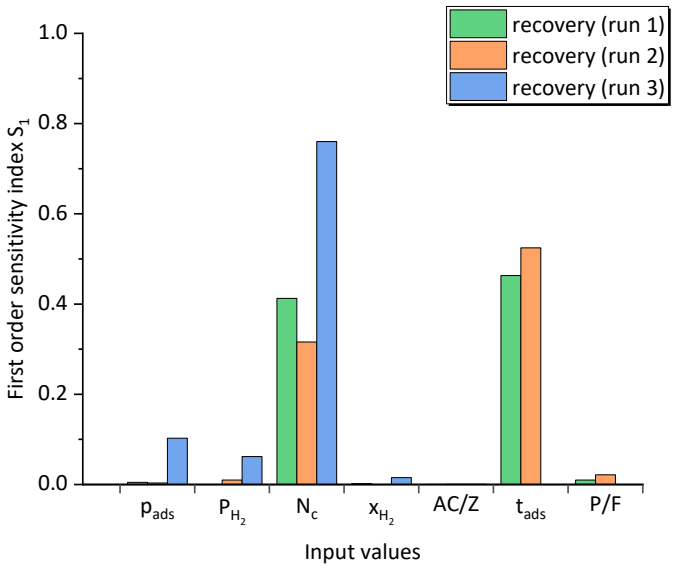

Fig. 8. First order sensitivity index S_i of hydrogen recovery for seven different input values and three different runs.

With increasing bed number, the hydrogen recovery increases, as **Fig. 9** illustrates. The highest increase of the recovery was observed at low bed numbers, e.g. from two to four beds, for all three cases. The sensitivity for run 3 is higher because the adsorption time is defined as an output value and thus cannot influence the recovery. For run 1 and 2 the adsorption time shows the second highest sensitivity index of 0.46 and 0.52, respectively. As it can be concluded from **Fig. 9**, a higher adsorption time leads to higher hydrogen recovery, but the influence is not as strong as the number of beds, as the sensitivity indexes already implied.

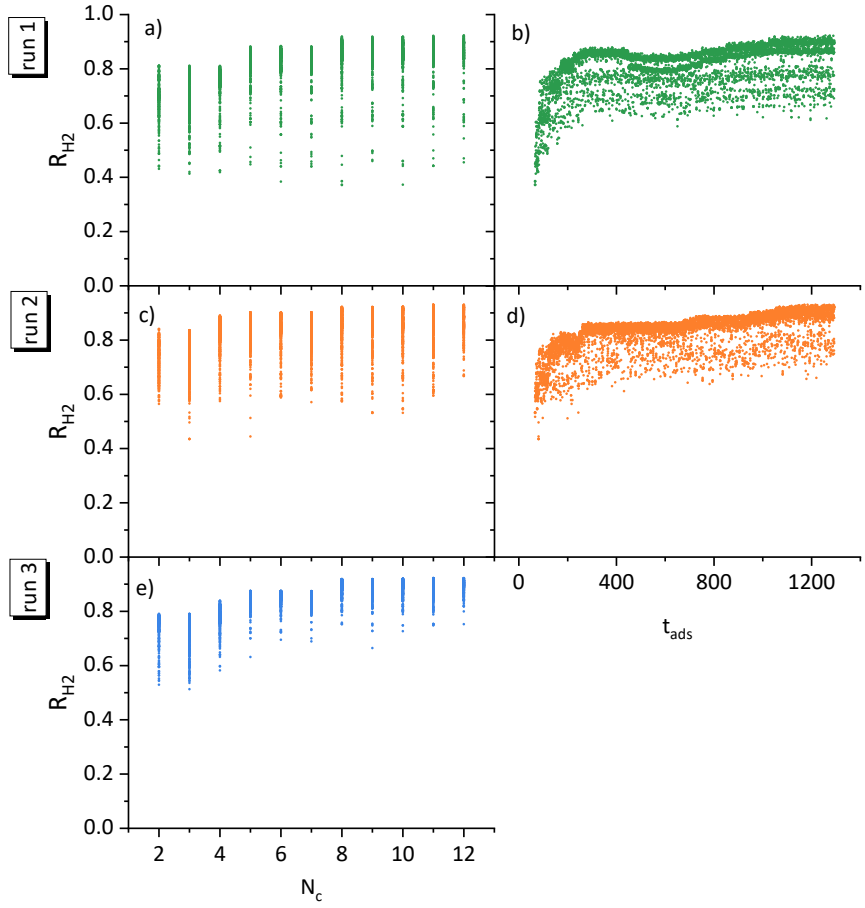

Fig. 9. Scattered hydrogen recovery values as a function of adsorption beds (N_c) and absorption time (t_{ads}) for three runs.

However, the five other input values have negligible to none influence on the hydrogen recovery. This was an unexpected result; the active carbon to zeolite ratio was expected to be of high importance to the hydrogen recovery because the type of adsorbent determines how strongly the different species are adsorbed. Only two of the seven scientific works that provided the data for this work included a systematical variation of the adsorbent ratio: Ahn et al. [18] and Moon et al. [23]. In both works the hydrogen recovery did not change significantly (less than 1 %). Therefore, the model trained in this work predicts only a very small influence either.

4.2 Hydrogen purity (run 1) and absorption time and P/F ratio (run 3)

The surrogate model predicts that the purity is mainly dependent on the hydrogen mole fraction in the feed, the purge to feed ratio, and the adsorption pressure, in descending order, see **Fig. 10**.

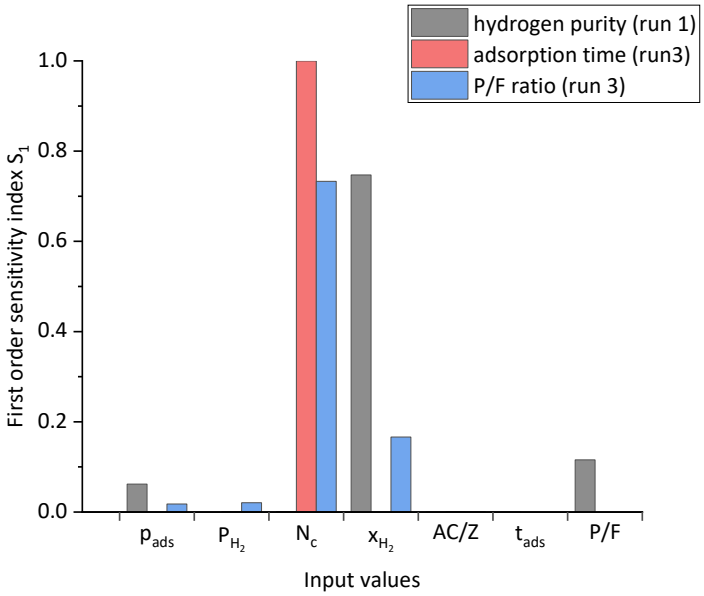

Fig. 10. First order sensitivity index S_i of hydrogen purity, adsorption time, and purge to feed ratio for seven different input values and run 1 and 3.

The corresponding S_1 values are 0.75, 0.12, and 0.06, respectively. Since there is no systematic change of the hydrogen mole fraction in the data, only seven different values were obtained and used to train the model. Therefore, the high sensitivity of the mole fraction on the purity must be investigated in more detail. **Fig. 11** illustrates the hydrogen purity, adsorption time, and purge to feed ratio as a function of the most important input parameters discussed before.

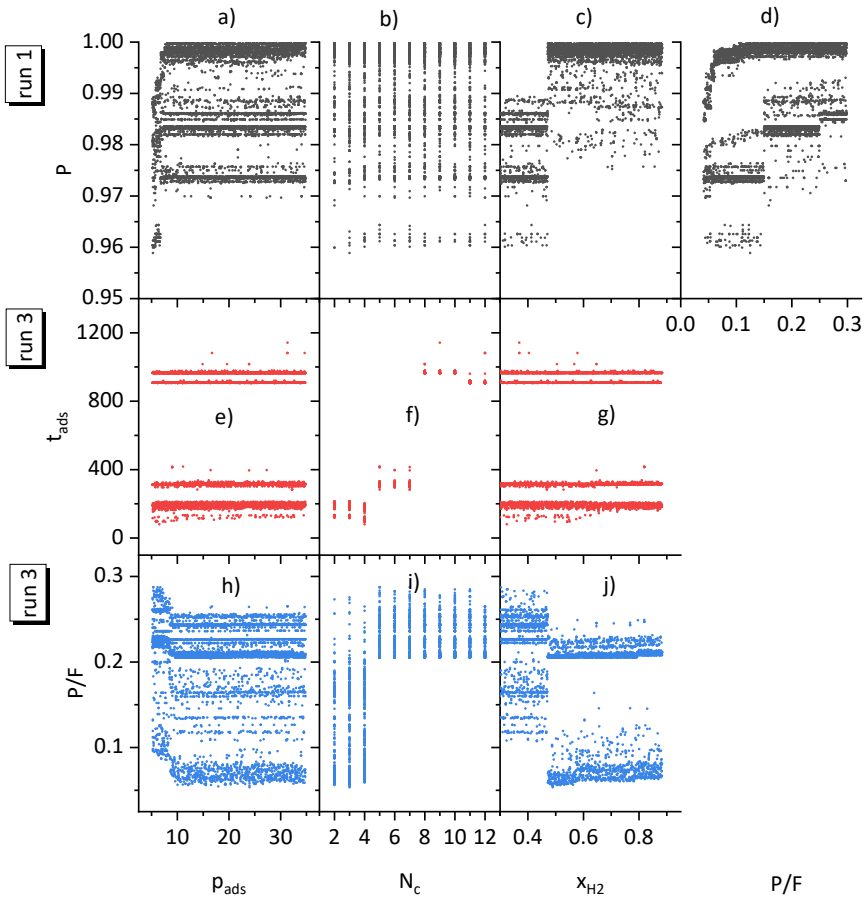

Fig. 11. Scattered hydrogen purity, adsorption time, and purge to feed ratio values as a function of adsorption pressure, number of adsorption beds, hydrogen mole fraction in the feed, and purge to feed flow ratio for run 1 and 3.

Fig. 11c shows an increase of the hydrogen purity with increasing feed mole fraction, but this increase is likely a step function from a mole fraction of 0.47, which is exactly located between two data sets with mole fractions of 0.38 and 0.564. Consequently, it is rather unlikely that this behavior is a true dependency and should be investigated experimentally in the future by varying the hydrogen feed mole fraction systematically. The adsorption pressure and the purge to feed ratio show a more credible dependency: with increasing adsorption pressure and purge to feed ratio the hydrogen purity also increases. This is consistent with the experimental results from Ahn et al. [18], for instance.

In run 3, the adsorption time and the purge to feed ratio were considered as output values to investigate their dependency on the other input values. The model predicts a solely dependency of the adsorption time on the number of adsorption beds ($S_1 = 1.0$). This is supported by the results of the sensitivity analysis shown in **Fig. 11**e-g. It

remains unclear, if the number of adsorption beds inherently increases the adsorption time, or if the trivial reason is that with smaller adsorption times, the recovery would decrease significantly, even with increasing bed numbers, and therefore it is always adjusted.

Finally, the purge to feed ratio shows a dependency on number of adsorption beds ($S_1 = 0.73$) and hydrogen mole fraction in the feed ($S_1 = 0.17$). Again, it remains unclear if this is a true dependency. **Fig. 11**j shows the same step for the hydrogen mole fraction as discussed before for the dependency of the purity on this value (**Fig. 11**c). Furthermore, the number of beds shows a similar step from 4 to 5 beds.

If the sensitivity analysis results and the training scores for run 3 are considered, it can be concluded that the combination of input and output values used for run 3 is no feasible choice to describe the investigated pressure swing adsorption process, at least with the data used in this study. On the contrary, for run 1 and run 2, the surrogate models predicted credible results.

5 Conclusions

A pressure swing adsorption surrogate model for hydrogen separation from nitrogen-diluted, fuel-rich operated HCCI engine exhaust gas mixtures was developed. Therefore, five different machine learning algorithms were used for training and testing 90 data sets from seven works on hydrogen separating PSA processes. Furthermore, the choice of input and output values for this surrogate model was discussed. The evaluation was performed by calculating the coefficient of determination R^2, mean absolute error MAE, and root mean squared error RMSE for a systematically, step-wise changed train size and for a five-fold cross-validation. Eventually, decision tree and random forest regression performed best in all cases and provided the highest R^2 and lowest MAE and RMSE values. A train size of 0.8 was found to be of a reasonable trade-off between test score and total score. At this point, the predicted hydrogen recovery for the random forest regression model showed a mean deviation of 6.6 % from the true value of the data sets.

Eventually, the random forest regression model and a train size of 0.8 were chosen for a global sensitivity analysis. This analysis revealed that the hydrogen recovery depends strongly on the number of adsorption beds and the adsorption time. The hydrogen purity depends on the adsorption pressure and the purge to feed ratio.

This study demonstrated the feasibility of creating surrogate models for complex, dynamic processes with a small number of data sets for model training. The accuracy of the results predicted by the models should nevertheless be treated with caution and only used for a first check of the meaningfulness of the examined model in a superordinate process concept. For increasing accuracy, a hyperparameter optimization could be performed in the future and/or more data sets should be used for training, as soon as more publications become available.

Beyond the specific results for PSA, the procedure for the selection of machine learning algorithms, their evaluation and application, including the sensitivity analysis, seems transferable to other problems, where complex systems shall be pre-evaluated without getting too deeply involved in physico-chemical modeling.

6 Acknowledgments

This research was funded by Deutsche Forschungsgemeinschaft (DFG), grant number 229243862 (AT24/13-3) within the framework of the DFG research unit FOR 1993 'Multi-functional conversion of chemical species and energy'. The authors gratefully acknowledge the DFG for the financial support.

7 Appendix

Table A1. Data sets from the literature that were the basis for the machine learning model training in this work. The productivity $Prod_{H2}$ determines the amount of hydrogen in mol which is separated per kg adsorbent per day and was not used for model training.

p_{ads}	p_{des}	R_{H2}	P_{H2}	T_F	N_c	x_{H2}	P/F	AC/Z	t_{ads}	$Prod_{H2}$	Ref.
bar	bar	-	-	°C	-	-	-	-	s	mol/kg/day	-
6.5	1	0.8636	0.9663	25.00	4	0.38	0.1	2.33	180		[18]
6.5	1	0.7931	0.9829	25.00	4	0.38	0.2	2.33	180		
6.5	1	0.7124	0.9943	25.00	4	0.38	0.3	2.33	180		
5	1	0.8121	0.9763	25.00	4	0.38	0.2	2.33	180		
6.5	1	0.7931	0.9829	25.00	4	0.38	0.2	2.33	180		
8	1	0.7738	0.9904	25.00	4	0.38	0.2	2.33	180		
8	1	0.7241	0.9881	25.00	4	0.38	0.2	2.33	160		
8	1	0.7931	0.9829	25.00	4	0.38	0.2	2.33	180		
8	1	0.8057	0.9789	25.00	4	0.38	0.2	2.33	200		
6.5	1	0.7982	0.9725	25.00	4	0.38	0.2	4.00	180		
6.5	1	0.7931	0.9829	25.00	4	0.38	0.2	2.33	180		
6.5	1	0.7909	0.9859	25.00	4	0.38	0.2	1.50	180		
34	1	0.7509	0.99976	30.00	4	0.8875	0.1	0.00	200	168.06	[19]
34	1	0.7268	0.9995	30.00	4	0.8875	0.2	0.00	200	162.67	
34	1	0.7056	0.99999	30.00	4	0.8875	0.3	0.00	200	157.93	
34	1	0.7875	0.99999	30.00	6	0.8875	0.3	0.00	600	117.51	
34	1	0.8198	0.99996	30.00	6	0.8875	0.3	0.00	700	122.33	
34	1	0.8441	0.99994	30.00	6	0.8875	0.3	0.00	800	125.95	
34	1	0.8629	0.99975	30.00	6	0.8875	0.3	0.00	900	128.75	
34	1	0.7611	0.99999	30.00	6	0.8875	0.3	0.00	150	113.57	
34	1	0.8247	0.9997	30.00	6	0.8875	0.3	0.00	200	123.05	
34	1	0.8626	0.9994	30.00	6	0.8875	0.3	0.00	250	128.71	
34	1	0.8879	0.9969	30.00	6	0.8875	0.3	0.00	300	132.48	
34	1	0.8705	0.99998	30.00	9	0.8875	0.3	0.00	800	86.64	
34	1	0.8994	0.99996	30.00	9	0.8875	0.3	0.00	1000	89.51	
34	1	0.9185	0.99993	30.00	9	0.8875	0.3	0.00	1200	91.41	

34	1	0.9258	0.99974	30.00	9	0.8875	0.3	0.00	1300	92.14	
34	1	0.8911	0.99999	30.00	12	0.8875	0.3	0.00	750	66.52	
34	1	0.9117	0.99996	30.00	12	0.8875	0.3	0.00	900	68.06	
34	1	0.9274	0.99993	30.00	12	0.8875	0.3	0.00	1050	69.15	
34	1	0.9383	0.99978	30.00	12	0.8875	0.3	0.00	1200	70.04	
5	1	0.6555	0.96355	25.00	2	0.38	0.2	2.33	200	75.23	[8]
6.5	1	0.6079	0.97536	25.00	2	0.38	0.2	2.33	200	78.92	
8	1	0.5760	0.99135	25.00	2	0.38	0.2	2.33	200	81.29	
6.5	1	0.7282	0.95467	25.00	2	0.38	0.1	2.33	200	94.67	
6.5	1	0.6079	0.97536	25.00	2	0.38	0.2	2.33	200	78.92	
6.5	1	0.4659	0.98787	25.00	2	0.38	0.3	2.33	200	60.74	
6.5	1	0.6079	0.97536	25.00	2	0.38	0.2	2.33	200	78.92	
6.5	1	0.5974	0.97696	25.00	2	0.38	0.2	2.33	200	74.68	
6.5	1	0.5930	0.97835	25.00	2	0.38	0.2	2.33	200	70.31	
6.5	1	0.5821	0.97966	25.00	2	0.38	0.2	2.33	200	66.21	
6.5	1	0.5920	0.9752	25.00	2	0.38	0.2	2.33	200	63.44	
6.5	1	0.5504	0.9814	25.00	2	0.38	0.2	2.33	180	75.91	
6.5	1	0.6079	0.97536	25.00	2	0.38	0.2	2.33	200	78.92	
6.5	1	0.6354	0.96918	25.00	2	0.38	0.2	2.33	220	83.12	
7	1	0.7140	0.9951	25.00	4	0.58	0.11	0.00	90	391.60	[20]
8	1	0.4330	0.9999	25.00	4	0.58	0.2	0.00	75	237.70	
8	1	0.4320	0.9998	25.00	4	0.58	0.21	0.00	75	237.10	
9	1	0.3720	0.9998	25.00	4	0.58	0.2	0.00	75	203.80	
8	1	0.2810	0.9998	25.00	4	0.58	0.23	0.00	60	154.20	
7	1	0.4940	0.9998	25.00	4	0.58	0.15	0.00	60	271.00	
8	1	0.5090	0.9998	25.00	4	0.58	0.14	0.00	75	279.20	
8	1	0.5030	0.9998	25.00	4	0.58	0.14	0.00	75	275.80	
8	1	0.6210	0.9984	25.00	4	0.58	0.09	0.00	75	340.70	
9	1	0.6160	0.9989	25.00	4	0.58	0.09	0.00	90	337.10	
9	1	0.6070	0.9988	25.00	4	0.58	0.1	0.00	90	333.40	
7	1	0.6060	0.9982	25.00	4	0.58	0.09	0.00	60	334.00	
8	1	0.5620	0.9992	25.00	4	0.58	0.18	0.00	90	308.10	
7	1	0.5490	0.9996	25.00	4	0.58	0.19	0.00	75	301.40	
9	1	0.3740	0.9998	25.00	4	0.58	0.13	0.00	60	205.30	
9	1	0.2970	0.9998	25.00	4	0.58	0.16	0.00	60	162.70	
9	1	0.4000	0.9997	25.00	4	0.58	0.16	0.00	90	219.60	
7	1	0.7540	0.9912	25.00	4	0.58	0.09	0.00	90	413.60	
9	1	0.5250	0.9999	25.00	4	0.58	0.1	0.00	75	283.10	
9	1	0.5590	0.9993	25.00	4	0.58	0.08	0.00	75	306.60	
8	1	0.7950	0.9827	25.00	4	0.58	0.04	0.00	90	436.00	

8	1	0.6810	0.9949	25.00	4	0.58	0.06	0.00	75	373.50	
9	1	0.3090	0.9998	25.00	4	0.58	0.17	0.00	60	170.00	
9	1	0.5140	0.9998	25.00	4	0.58	0.16	0.00	90	282.00	
9	1	0.7600	0.9999	25.00	2	0.7	0.11	0.00	180		[21]
11	1	0.7400	0.9999	25.00	2	0.7	0.11	0.00	180		
13	1	0.6800	0.9999	25.00	2	0.7	0.11	0.00	180		
16	1	0.6600	0.9999	25.00	2	0.7	0.11	0.00	180		
11	1	0.8200	0.96	25.00	2	0.7	0.05	0.00	180		
11	1	0.6600	0.9999	25.00	2	0.7	0.175	0.00	180		
10	1	0.5773	0.9996	25.00	2	0.564	0.11	0.65	168	152.16	[22]
10	1	0.7921	0.9957	25.00	2	0.564	0.05	0.65	200	186.24	
10	1	0.7167	0.9952	25.00	2	0.564	0.05	0.65	180	199.20	
10	1	0.7254	0.9993	25.00	2	0.564	0.07	0.65	170	168.48	
10	1	0.7890	0.9954	25.00	2	0.564	0.05	0.65	200	191.52	
10	1	0.6627	0.9993	25.00	2	0.564	0.08	0.65	167	172.08	
10	1	0.7509	0.9985	25.00	2	0.564	0.06	0.65	175	179.04	
25	1.1	0.7699	0.9951	35.00	2	0.88	0.1	1.00	170	540.40	[23]
30	1.1	0.7491	0.9972	35.00	2	0.88	0.1	1.00	170	535.80	
35	1.1	0.7304	0.9978	35.00	2	0.88	0.1	1.00	170	530.1	
25.00	1.1	0.7764	0.9977	35.00	2	0.88	0.1	4.00	170	545.4	
30.00	1.1	0.7505	0.9993	35.00	2	0.88	0.1	4.00	170	550	
35.00	1.1	0.7330	0.9995	35.00	2	0.88	0.1	4.00	170	555	
35	1.1	0.7736	0.9972	35.00	2	0.88	0.05	4.00	170	585.8	
35	1.1	0.6554	0.9996	35.00	2	0.88	0.2	4.00	170	496.4	

References

[1] N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, P. Kevin Tucker, Surrogate-based analysis and optimization, Progress in Aerospace Sciences 41 (2005) 1–28. https://doi.org/10.1016/j.paerosci.2005.02.001.

[2] K. Banke, R. Hegner, D. Schröder, C. Schulz, B. Atakan, S.A. Kaiser, Power and syngas production from partial oxidation of fuel-rich methane/DME mixtures in an HCCI engine, Fuel 243 (2019) 97–103. https://doi.org/10.1016/j.fuel.2019.01.076.

[3] D. Schröder, K. Banke, S.A. Kaiser, B. Atakan, The kinetics of methane ignition in fuel-rich HCCI engines: DME replacement by ozone, Proceedings of the Combustion Institute (2020). https://doi.org/10.1016/j.proci.2020.05.046.

[4] D. Schröder, R. Hegner, A. Güngör, B. Atakan, Exergoeconomic analysis of an HCCI engine polygeneration process, Energy Conversion and Management 203 (2020) 112085. https://doi.org/10.1016/j.enconman.2019.112085.

[5] X. Zhu, S. Li, Y. Shi, N. Cai, Recent advances in elevated-temperature pressure swing adsorption for carbon capture and hydrogen production, Progress in Energy and Combustion Science 75 (2019) 100784. https://doi.org/10.1016/j.pecs.2019.100784.
[6] H.R. Sant Anna, A.G. Barreto, F.W. Tavares, M.B. de Souza, Machine learning model and optimization of a PSA unit for methane-nitrogen separation, Computers & Chemical Engineering 104 (2017) 377–391. https://doi.org/10.1016/j.compchemeng.2017.05.006.
[7] F. Ye, S. Ma, L. Tong, J. Xiao, P. Bénard, R. Chahine, Artificial neural network based optimization for hydrogen purification performance of pressure swing adsorption, International Journal of Hydrogen Energy 44 (2019) 5334–5344. https://doi.org/10.1016/j.ijhydene.2018.08.104.
[8] J. Xiao, C. Li, L. Fang, P. Böwer, M. Wark, P. Bénard, R. Chahine, Machine learning–based optimization for hydrogen purification performance of layered bed pressure swing adsorption, Int J Energy Res 44 (2020) 4475–4492. https://doi.org/10.1002/er.5225.
[9] L. Tong, P. Bénard, Y. Zong, R. Chahine, K. Liu, J. Xiao, Artificial neural network based optimization of a six-step two-bed pressure swing adsorption system for hydrogen purification, Energy and AI 5 (2021) 100075. https://doi.org/10.1016/j.egyai.2021.100075.
[10] K.T. Leperi, D. Yancy-Caballero, R.Q. Snurr, F. You, 110th Anniversary Surrogate Models Based on Artificial Neural Networks To Simulate and Optimize Pressure Swing Adsorption Cycles for CO 2 Capture, Ind. Eng. Chem. Res. 58 (2019) 18241–18252. https://doi.org/10.1021/acs.iecr.9b02383.
[11] S.G. Subraveti, Z. Li, V. Prasad, A. Rajendran, Machine Learning-Based Multiobjective Optimization of Pressure Swing Adsorption, Ind. Eng. Chem. Res. 58 (2019) 20412–20422. https://doi.org/10.1021/acs.iecr.9b04173.
[12] C.M. Rebello, P.H. Marrocos, E.A. Costa, V.V. Santana, A.E. Rodrigues, A.M. Ribeiro, I.B.R. Nogueira, Machine Learning-Based Dynamic Modeling for Process Engineering Applications: A Guideline for Simulation and Prediction from Perceptron to Deep Learning, Processes 10 (2022) 250. https://doi.org/10.3390/pr10020250.
[13] K.N. Pai, V. Prasad, A. Rajendran, Experimentally validated machine learning frameworks for accelerated prediction of cyclic steady state and optimization of pressure swing adsorption processes, Separation and Purification Technology 241 (2020) 116651. https://doi.org/10.1016/j.seppur.2020.116651.
[14] Z. Hao, C. Zhang, A.A. Lapkin, Efficient surrogates construction of chemical processes: Case studies on pressure swing adsorption and gas-to-liquids, AIChE Journal (2022). https://doi.org/10.1002/aic.17616.
[15] Y. Yan, Z. Shi, H. Li, L. Li, X. Yang, S. Li, H. Liang, Z. Qiao, Machine learning and in-silico screening of metal–organic frameworks for O2/N2 dynamic adsorption and separation, Chemical Engineering Journal 427 (2022) 131604. https://doi.org/10.1016/j.cej.2021.131604.
[16] M. Luberti, H. Ahn, Review of Polybed pressure swing adsorption for hydrogen purification, International Journal of Hydrogen Energy 47 (2022) 10911–10933. https://doi.org/10.1016/j.ijhydene.2022.01.147.
[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D.

Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[18] S. Ahn, Y.-W. You, D.-G. Lee, K.-H. Kim, M. Oh, C.-H. Lee, Layered two- and four-bed PSA processes for H2 recovery from coal gas, Chemical Engineering Science 68 (2012) 413–423. https://doi.org/10.1016/j.ces.2011.09.053.

[19] M. Luberti, D. Friedrich, S. Brandani, H. Ahn, Design of a H2 PSA for cogeneration of ultrapure hydrogen and power at an advanced integrated gasification combined cycle with pre-combustion capture, Adsorption 20 (2014) 511–524. https://doi.org/10.1007/s10450-013-9598-0.

[20] M. Yáñez, F. Relvas, A. Ortiz, D. Gorri, A. Mendes, I. Ortiz, PSA purification of waste hydrogen from ammonia plants to fuel cell grade, Separation and Purification Technology 240 (2020) 116334. https://doi.org/10.1016/j.seppur.2019.116334.

[21] J. Yang, C.-H. Lee, J.-W. Chang, Separation of Hydrogen Mixtures by a Two-Bed Pressure Swing Adsorption Process Using Zeolite 5A, Ind. Eng. Chem. Res. 36 (1997) 2789–2798. https://doi.org/10.1021/ie960728h.

[22] N. Zhang, P. Bénard, R. Chahine, T. Yang, J. Xiao, Optimization of pressure swing adsorption for hydrogen purification based on Box-Behnken design method, International Journal of Hydrogen Energy 46 (2021) 5403–5417. https://doi.org/10.1016/j.ijhydene.2020.11.045.

[23] D.-K. Moon, D.-G. Lee, C.-H. Lee, H2 pressure swing adsorption for high pressure syngas from an integrated gasification combined cycle with a carbon capture process, Applied Energy 183 (2016) 760–774. https://doi.org/10.1016/j.apenergy.2016.09.038.

[24] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning, Springer New York, New York, NY, 2013.

[25] Jeff Reback, jbrockmendel, Wes McKinney, Joris Van den Bossche, Tom Augspurger, Phillip Cloud, Simon Hawkins, Matthew Roeschke, gfyoung, Sinhrks, Adam Klein, Patrick Hoefler, Terji Petersen, Jeff Tratner, Chang She, William Ayd, Shahar Naveh, JHM Darbyshire, Marc Garcia, Richard Shadrach, Jeremy Schendel, Andy Hayden, Daniel Saxton, Marco Edward Gorelli, Fangchen Li, Matthew Zeitlin, Vytautas Jancauskas, Ali McMaster, Pietro Battiston, Skipper Seabold, pandas-dev/pandas: Pandas 1.3.4, Zenodo, 2022.

[26] B. Schenker, M. Agarwal, Cross-validated structure selection for neural networks, Computers & Chemical Engineering 20 (1996) 175–186. https://doi.org/10.1016/0098-1354(95)00013-R.

[27] O. Kramer, K-Nearest Neighbors, in: O. Kramer (Ed.), Dimensionality Reduction with Unsupervised Nearest Neighbors, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 13–23.

[28] D.J. Denis, Applied Univariate, Bivariate, and Multivariate Statistics Using Python, Wiley, 2021.

[29] C.J. Willmott, K. Matsuura, Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance, Clim. Res. 30 (2005) 79–82. https://doi.org/10.3354/cr030079.

[30] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, second ed., Springer New York, New York, NY, 2009.

[31] P. Jiang, Surrogate Model-Based Engineering Design and Optimization, Springer Singapore Pte. Limited, Singapore, 2020.
[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Epsilon-Support Vector Regression. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html (accessed 17 March 2022).
[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, A decision tree regressor. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html (accessed 17 March 2022).
[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, A random forest regressor. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html (accessed 17 March 2022).
[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Multi-layer Perceptron regressor. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html (accessed 17 March 2022).
[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Regression based on k-nearest neighbors. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html (accessed 17 March 2022).
[37] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and Computing 14 (2004) 199–222. https://doi.org/10.1023/B%3ASTCO.0000035301.49549.88.
[38] N. Cristianini, J. Shawe-Taylor, An introduction to support vector machines and other kernel-based learning methods, Cambridge Univ. Press, Cambridge, 2000.
[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Epsilon-Support Vector Regression Kernel Functions. https://scikit-learn.org/stable/modules/svm.html#kernel-functions (accessed 17 March 2022).
[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Multi target regression. https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputRegressor.html (accessed 17 March 2022).
[41] Y.-Y. Song, Y. Lu, Decision tree methods: applications for classification and prediction, Shanghai Arch. Psychiatry 27 (2015) 130–135. https://doi.org/10.11919/j.issn.1002-0829.215044.
[42] W. Usher, J. Herman, C. Whealton, D. Hadka, xantares, F. Rios, bernardoct, C. Mutel, J. van Engelen, Salib/Salib: Launch!, Zenodo, 2016.
[43] A. Saltelli, M. Ratto, T. Andres, Global Sensitivity Analysis. The Primer, 2008.

[44] D. Schröder, R. Hegner, A. Güngör, B. Atakan, The influence of uncertainty of economic parameters and upscaling on product costs of an engine polygeneration system, in: ECOS 2019 : Proceedings of the 32nd International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems, Wrocław, Poland, 2019, pp. 1455–1466.