

Machine Learning Applications

Winter semester 2019/2020

Henrik Simon & Sebastian Baumann

Lecture VIII Diagnostics

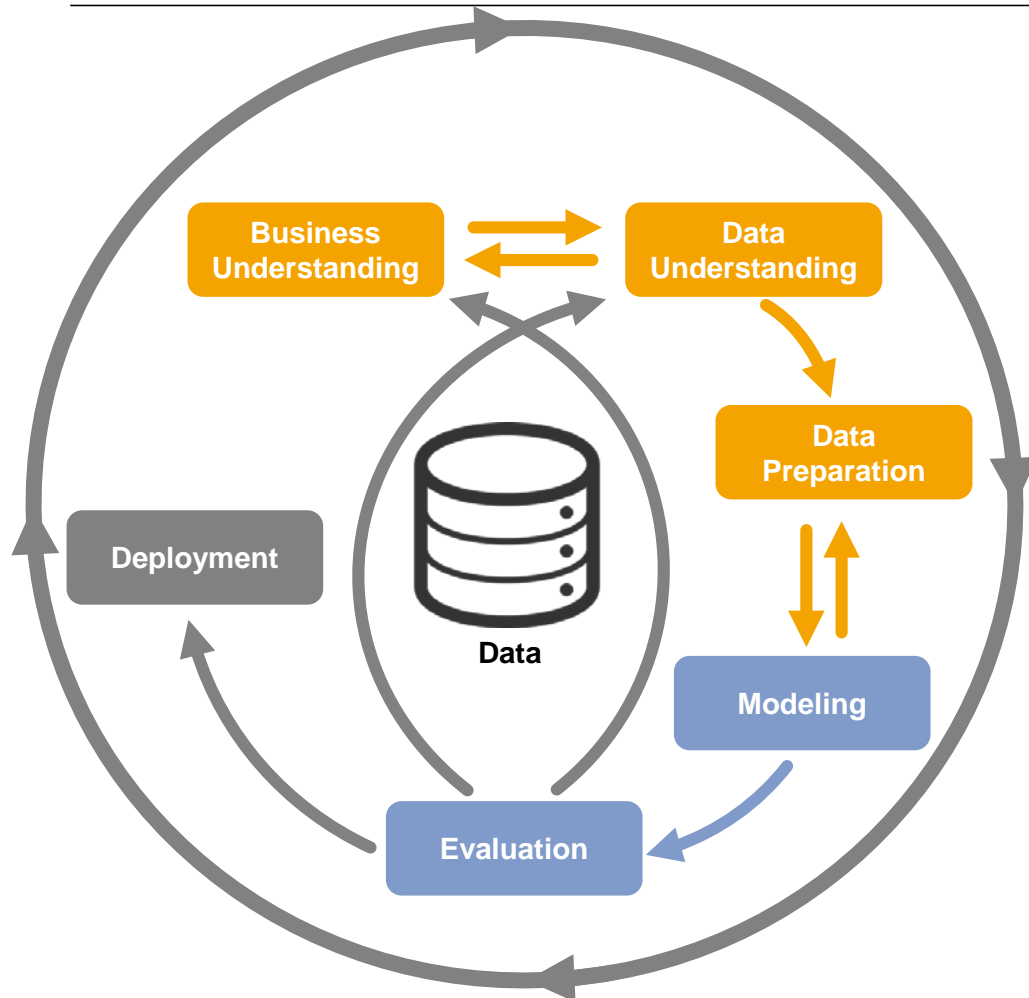


What should you be able to take out of the lecture today?

- **CRISP-DM:** Foundations of modeling, evaluation and deployment phases
- **Data mining:** association analysis with apriori algorithms
- **Development of model understanding:**
 - comparison of different machine learning models on an exemplary data set
 - Deeper insights into how neural networks and decision trees / random forests work and how they are set up
 - Models to assess and diagnose aircraft fuel flow
 - Illustration of insufficient data quality: assessment and identification of underfitting and overfitting
- **Model assessment:**
 - overview of common evaluation metrics and their application to two specific application cases from aviation / PHM
 - Discussion of advantages and disadvantages of the model performance metrics
- **Evaluation:** assessment of business case advantages

RECAP: PROCESS MODELS AND MACHINE LEARNING APPROACH

The hard work begins to pay off with modeling.

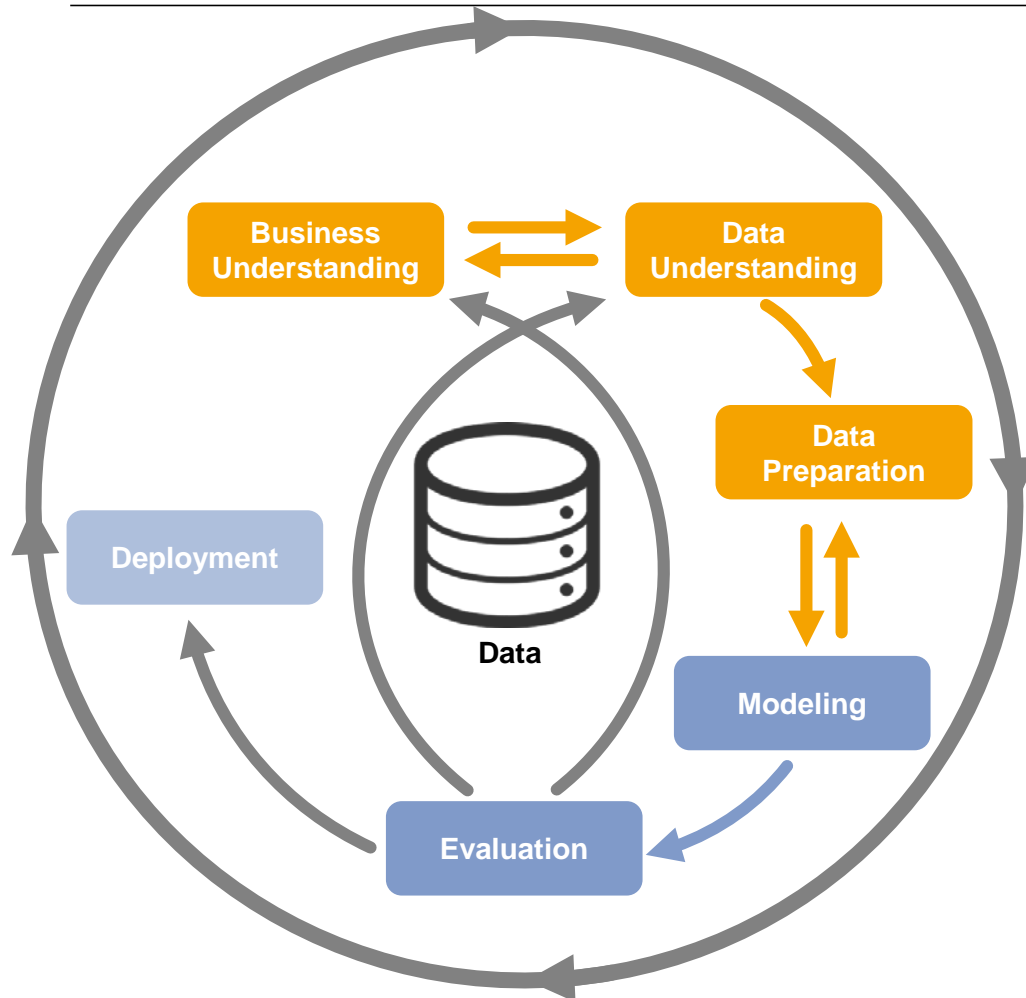


Modeling

- **Select modeling technique** (possibly already defined during the business understanding phase). If multiple techniques are applied, the task must be run through separately for each technique, usually determination of several models and benchmarking. Consideration of possible model assumptions (e. g. uniform distributions of all attributes, no missing values, class attribute must be symbolic etc.). Application of suitable data mining methods, optimization of parameters.
- **Generate test design** (procedure or mechanism) to test the model's quality and validity (e. g. error rates as quality measures etc.). Determining a suitable division and separating the dataset into train and test sets
- **Build model(s)** on the train set, and estimate its (their) quality on the separate test set. Finding an optimal setting of adjusting parameters. Save produced models and model descriptions.
- **Assess model(s)** according to existing domain knowledge using data mining success criteria and desired test design. Judge the success of the application of modeling and discovery techniques technically, discussion of the results in the task and business context and ranking of models.
- **Revising and tuning of model parameters** according to the model assessment. Iterate model building and assessment until finding best model(s) (accuracy, precision, robustness, explainability, generalizability)

Sources: https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.crispdm.help/crisp_deployment_phase.htm | <https://www.sv-europe.com/crisp-dm-methodology/#modeling>

Reviewing business success criteria and making improvements within an organization complete the cycle.



Evaluation

- **Select the model that best suits the task at hand.** Careful comparison with the task at hand.
- **Assess the level of achievement of the business objective** to determine if there is a business reason why this model is inadequate.
- **Test models on test applications.** A generated model that meet the selected criteria best becomes an approved model.
- **Review results for quality assurance questions:** Creating Models Correctly, only with attributes that can be used and will be available for future analysis?
- **Listing of future actions and decisions**

Deployment

- **Preparation and presentation of the results, record general procedures, prepare final report and monitoring reportings**
- **Plan deployment and integration of the model(s)** into the client's decision-making process where appropriate: summarize deployment strategy and necessary performing steps
- **Plan monitoring and maintenance:** day-to-day businesses to avoid (periods of) incorrect usage of the results
- **Review project:** experience documentation with dos and don'ts, learnings and best practices

Sources: https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.crispdm.help/crisp_deployment_phase.htm | <https://www.sv-europe.com/crisp-dm-methodology/#modeling>

DIAGNOSIS VS. PROGNOSIS

The different capabilities provide a distinction between diagnosis and prognosis.

Ops

Description

Diagnosis

Prognosis

Prescription

Acquisition

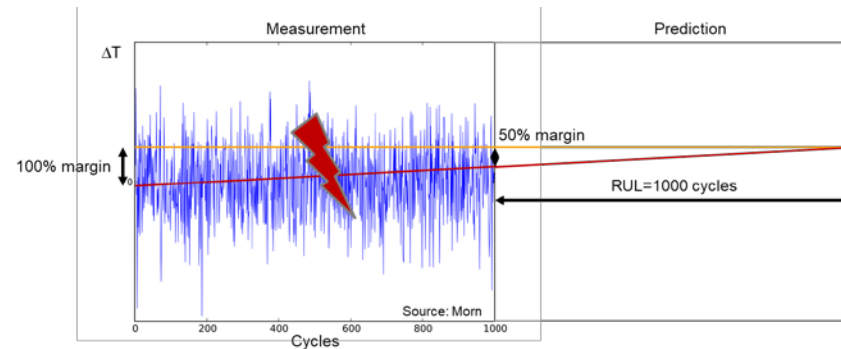
Understanding & wrangling

Use of descriptive findings and models to describe a system state

Behavioral estimation on the basis of the current state and future influences

Derivation of decision support and recommendations / automation

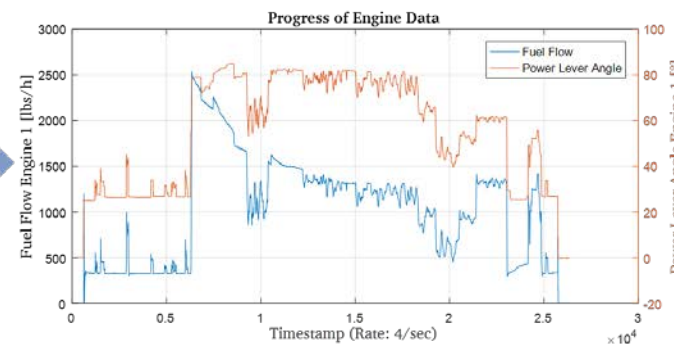
An example from
rail transport



An example from
aviation



Source: karlenepetit.blogspot.com



Predictable predictor?
Predictive capability?

DATA AND MODEL QUALITY DEPENDENCIES

Bias-Variance-Trade-Off

- **Underfitting**

Model is unable to capture the global behavior or pattern of the data. Possible causes: less amount of data, linear modeling with nonlinear data.

- **Overfitting**

Model complexity approaches the complexity of training data, e. g. captures noise in data.

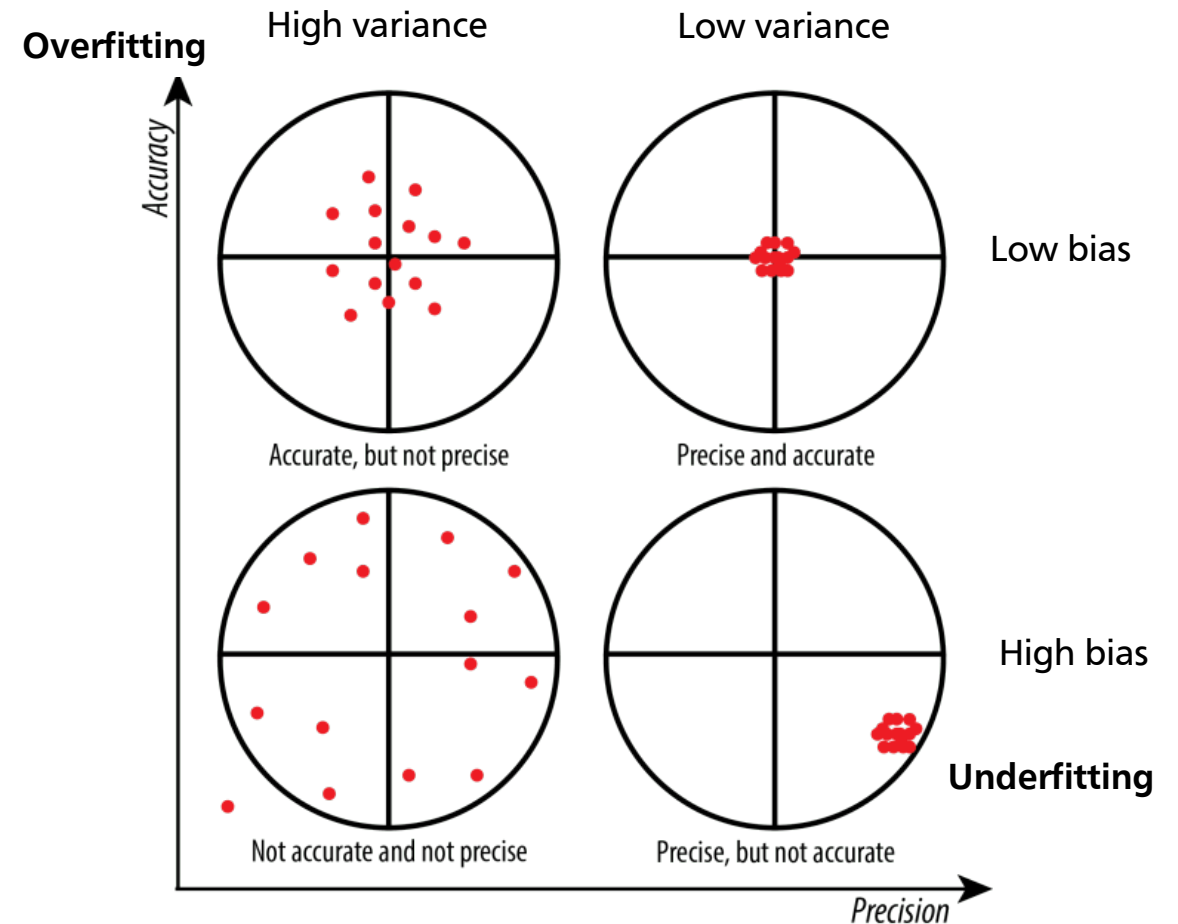
- **Bias-Variance-Trade-Off**

mean approximation of the data

- simple models: bias to generalized data behavior
- complex models: Variance increases on test data

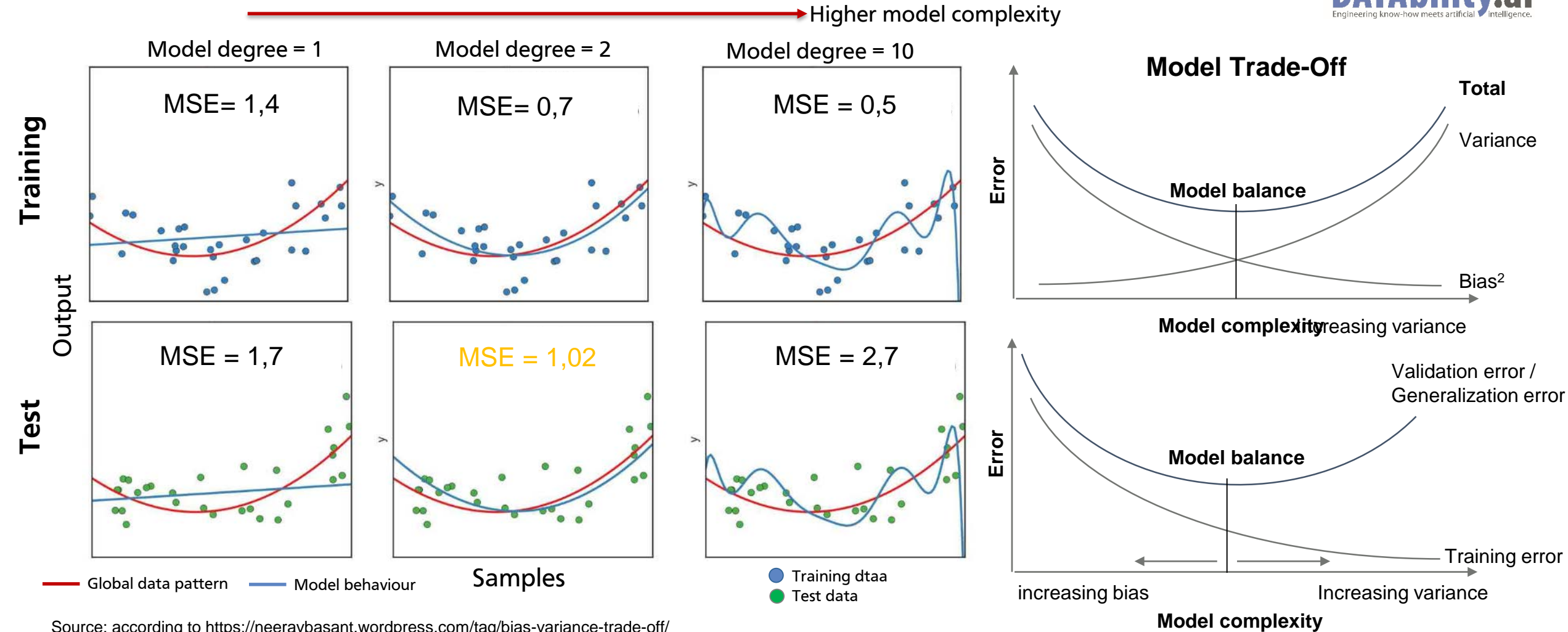
- **Generalizability**

A model is over adjusted by agent A, if agent A* describes the training data worse with a larger error but the overall distribution of the data with a smaller error better than A.



Source: according to <https://wp.stolaf.edu/it/gis-precision-accuracy/>

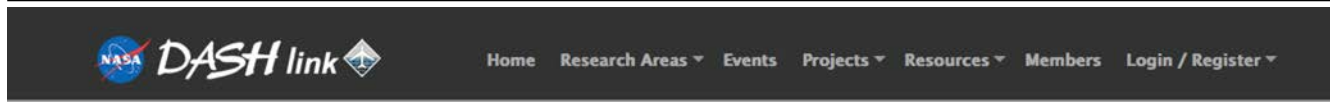
Achievement of generalizability: accuracy paradoxon



Source: according to <https://neeravbasant.wordpress.com/tag/bias-variance-trade-off/>

EXAMPLE DATA EXPLANATION

Sample full flight data



A web-based collaboration tool for those interested in data mining and systems health

LEARN MORE

Research Areas

Learn about our research fields, goals and their associated projects.

Projects

See what others in the community are working on. Join or start your own.

Resources

Available data sets, algorithms, and publications FREE to download

[<https://c3.ndc.nasa.gov/dashlink/>]

Available data volume

- approx. 200 GB (.zip)
(approx. 3 MB per flight)
- 186 attributes / parameters

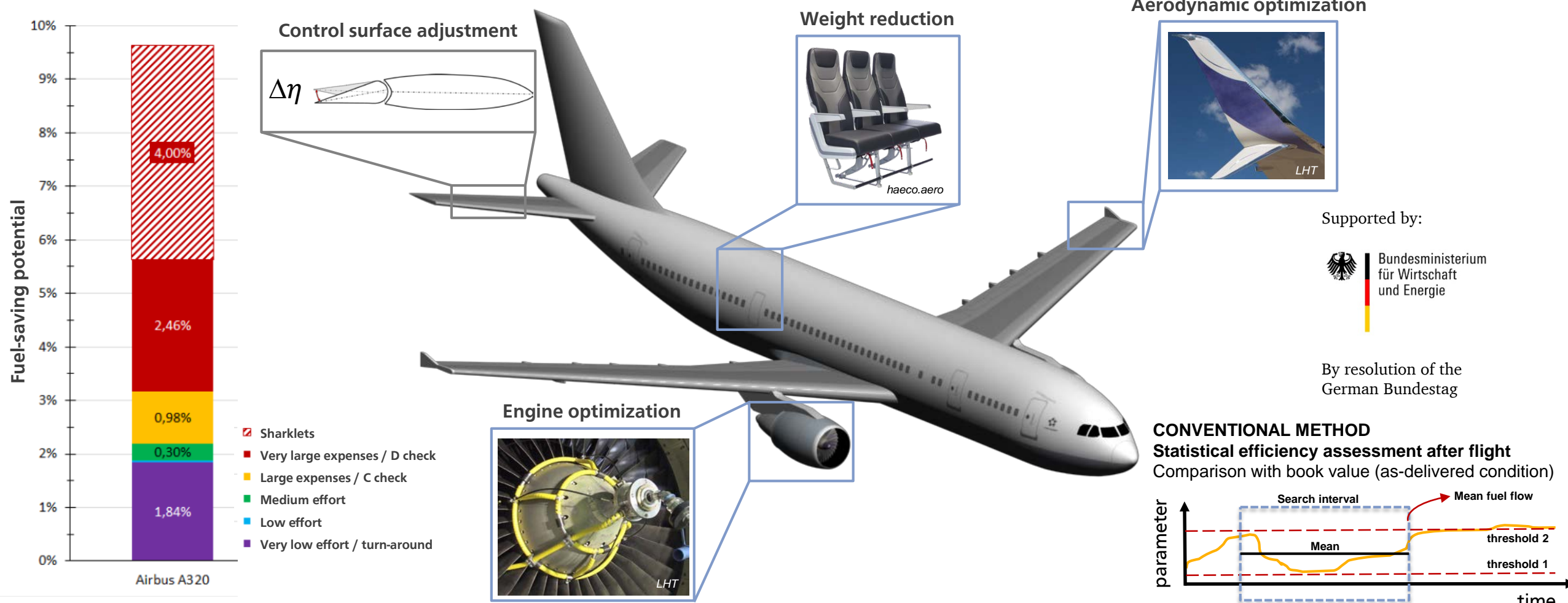
Available data cluster

- environmental data
- engine parameters
- navigation parameters
- pilot inputs
- system states: errors, warnings

Attribute excerpt

PARAMETER	DESCRIPTION	UNIT	RATE
AIL	AILERON POSITION LH	DEG	1/s
ALT	PRESSURE ALTITUDE	FEET	4/s
ALTR	ALTITUDE RATE	FT/MIN	4/s
AOA	ANGLE OF ATTACK	DEG	4/s
EGT	EXHAUST GAS TEMP.	DEG	4/s
FF	FUEL FLOW	LBS/HR	4/s
FQTY	FUEL QUANTITY TANK	LBS	1/s
LATP	LATITUDE POSITION	DEG	1/s
MACH	MACH	MACH	4/s
N1	FAN SPEED 1	%RPM	4/s
PLA	POWER LEVER ANGLE	DEG	4/s
PT	TOTAL PRESSURE	MB	2/s
PTCH	PITCH ANGLE	DEG	8/s
RUDD	RUDDER POSITION	DEG	2/s
TAS	TRUE AIRSPEED	KNOTS	4/s
TAT	TOTAL AIR TEMPERATURE	DEG	1/s
TH	TRUE HEADING	DEG	4/s
VIB	ENGINE VIBRATION	IN/SEC	4/s
WD	WIND DIRECTION	DEG	4/s
WS	WIND SPEED	KNOTS	4/s

Project motivation: Suitable validations for the effectiveness of retrofits at $\pm 0.X$ % efficiency variations are still lacking.



DATA MINING

Apriori algorithm is a method for association analysis in data mining.

Bottom up approach: algorithm finds all frequent itemsets with minimum support in a database. Frequent item subsets extended one item at a time (candidate generation), groups of candidates are tested against the data. Algorithm terminates when no further successful extensions are found. Then it prunes the candidates which have an infrequent sub pattern. The minimum confidence constraint are used to form rules.

Given

- R as a set of objects (item of a transaction)
- t as a transaction (rows),
- r as a set of transactions

find all rules c of the format $X \rightarrow Y$ with $X \subseteq R, Y \subseteq R, X \cap Y = \{\}$ for which

- a minimal support threshold s_{\min} (relative frequency) and
- a minimum confidence conf_{\min} (trustworthiness) applies.

Counting candidate item sets is efficient using breadth-first search and a Hash tree structure and downward-closure theorem (if a quantity is frequent, so are all its subsets) called anti-monotonicity. The complexity in time and space is given with $\mathcal{O}(2^R)$.

References:

R. Agrawal, R. Srikant: Fast Algorithms for Mining Association Rules in Large Databases. In: 20th International Conference on Very Large Data Bases, 478-499, 1994.
<https://arxiv.org/ftp/arxiv/papers/1406/1406.7371.pdf>

Exemplary results for association rules of association analyses with apriori algorithms on data sets

Shopping basket analysis

ID	Aftershave	Beer	Crisps
1	0	1	1
2	1	1	0
3	0	1	1
4	1	0	0

Results

{Aftershave} → {Beer} : s=0.25, conf=0.5
{Aftershave} → {Crisps} : s=0
{Beer} → {Crisps} : s=0.25, conf=0.66
{Crisps} → {Aftershave} : s=0
{Aftershave} → {Beer, Crisps} : s=0

➤ **Recommendation for offer adjustment**

Results of the apriori algorithm on given flight data

Item A	Instances	Item B	Confidence
ALT='(30570.7-inf)'	45311	PS='(-inf-9.14279]'	1
PS='(-inf-9.14279]'	48253	ALT='(30570.7-inf)'	0.94
ALT='(28109.4-30570.7]'	59337	PS='(9.14279-10.55748]'	0.95
PS='(9.14279-10.55748]'	105937	ALT='(28109.4-30570.7]'	0.53
ALT='(25648.1-28109.4]'	66666	PS='(9.14279-10.55748]'	0.74
PS='(9.14279-10.55748]'	105937	ALT='(25648.1-28109.4]'	0.47
PS='(9.14279-10.55748]'	105937	TAT='(-17.175-11.15]'	0.38
TAT='(-17.175-11.15]'	59795	PS='(9.14279-10.55748]'	0.67
PS='(9.14279-10.55748]'	105937	FF='(5028-5793.6]'	0.52
FF='(5028-5793.6]'	94905	PS='(9.14279-10.55748]'	0.59

Results

- Identification of the height-pressure and height-temperature relations
- **Identification of cruise flight altitude and consumption**

MODEL PERFORMANCE METRICS

Categorization of (prognostic) metrics based on end usage (1/2)

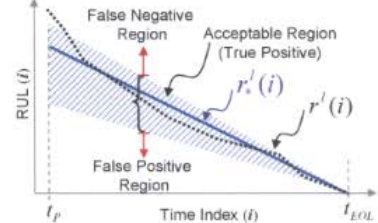
Metrics	Assessment Goals	Operations				Engineering		Regulatory
		Program Manager	Plant Manager	Operator	Maintainer	Designer	Researcher	Policy Maker
Certification Metrics	Assess conformance to safety assurance and certification requirements	X						X
Cost-Benefit Metrics	Assess the economic viability for specific applications before It can be approved or funded	X						X
Reliability Based Metrics	Assess probabilities of failures based on statistical evidence from multiple systems			X		X		X
Algorithm performance Metrics	Assess performance of prediction algorithms in predicting EoL	X	X	X	X		X	X
Computational Performance Metrics	Assess computational requirements					X	X	

Categorization of (prognostics) metrics based on end usage (2/2)

Category	End User	Goals	Metrics
Operations	Program Manager	Assess the economic viability of prognosis technology for specific applications before it can be approved and funded.	Cost-benefit type metrics that translate prognostics performance in terms of tangible and intangible cost savings.
	Plant Manager	Resource allocation and mission planning based on available prognostic information.	Accuracy and precision based metrics that compute RUL estimates for specific Unit Under Test (UUT). Such predictions are based on degradation or damage accumulation models.
	Operator	Take appropriate action and carry out re-planning in the event of contingency during mission.	Accuracy and precision based metrics that compute RUL estimates for specific UUTs. These predictions are based on fault growth models for critical failures.
	Maintainer	Plan maintenance in advance to reduce UUT downtime and maximize availability.	Accuracy and precision based metrics that compute RUL estimates based on damage accumulation models.

Category	End User	Goals	Metrics
Engineering	Designer	Implement the prognostic system within the constraints of user specifications. Improve performance by modifying design.	Reliability based metrics to evaluate a design and identify performance bottlenecks. Computational performance metrics to meet resource constraints.
	Researcher	Develop and Implement robust performance assessment algorithms with desired confidence levels.	Accuracy and Precision based metrics that employ uncertainty management and output probabilistic predictions in presence of uncertain conditions.
Regulatory	Policy Makers	To assess potential hazards (safety, economic, and social) and establish policies to minimize their effects.	Cost-benefit-risk measures, Accuracy and Precision based RUL measures to establish guidelines & timelines for phasing out of aging fleet and/or resource allocation for future projects.

Characteristics of different evaluation metrics for algorithmic performance (focus on PHM) (1/3)

Metric Name	Definition	Description	Range
Accuracy Based Metrics			
Error	$\Delta^i(i) = r_s^i(i) - r^i(i)$	Error defines the basic notion of deviation from desired output. Most accuracy based metrics are derived directly or indirectly from error.	$(-\infty, \infty)$ Perfect score = 0
Average scale independent error	$A(i) = \frac{1}{L} \sum_{i=1}^L \exp\left\{-\left \frac{\Delta^i(i)}{D_0}\right \right\}$	Weights exponentially the errors in RUL predictions and averages over several UUTs; where D_0 is a normalizing constant whose value depends on the magnitudes in the application.	$(0, 1]$ Perfect score = 1
Average bias	$B_i = \frac{\sum_{i=P}^{EOP} \{\Delta^i(i)\}}{(EOP - P + 1)}$	Averages the errors in predictions made at all subsequent times after prediction starts for the i^{th} UUT. This metric can be extended to average biases over all UUTs to establish overall bias.	$(-\infty, \infty)$ Perfect score = 0
Timeliness	$A(i) = \frac{1}{L} \sum_{i=1}^L \phi\{\Delta^i(i)\}$ where, $\phi(z) = \begin{cases} \exp\{z/a_1\} - 1, & \text{if } z < 0 \\ \exp\{z/a_2\} - 1, & \text{if } z \geq 0 \end{cases}$ and $a_1 > a_2 > 0$	Exponentially weighs RUL prediction errors through an asymmetric weighting function. Penalizes the late predictions more than early predictions.	$(0, \infty)$ Perfect score = 0
False Positives (FP)	$FP(r_s^i(i)) = \begin{cases} 1 & \text{if } \Delta^i(i) > t_{FP} \\ 0 & \text{otherwise} \end{cases}$ where t_{FP} = user defined acceptable early prediction	FP assesses unacceptable early predictions and FN assesses unacceptable late predictions at specified time instances. User must set acceptable ranges (t_{FN} and t_{FP}) for prediction. Early predictions result in excessive lead time, which may lead to unnecessary corrections. Also note that, a prediction that is late more than a critical threshold time units (t_c) is equivalent to not making any prediction and having the failure occurring.	$[0, 1]$ Perfect score = 0
False Negatives (FN)	$FN(r_s^i(i)) = \begin{cases} 1 & \text{if } -\Delta^i(i) > t_{FN} \\ 0 & \text{otherwise} \end{cases}$ where t_{FN} = user defined acceptable late prediction		$[0, 1]$ Perfect score = 0
Mean absolute percentage error (MAPE)	$MAPE(i) = \frac{1}{L} \sum_{i=1}^L \left \frac{100\Delta^i(i)}{r_s^i(i)} \right $	Averages the absolute percentage errors in the predictions of multiple UUTs at the same prediction horizon. Instead of the mean, median can be used to compute Median absolute percentage error (MdAPE) in a similar fashion.	$[0, \infty)$ Perfect score = 0

Source: Saxena, Celaya, Balaban, Goebel, Saha, Saha, Schwabacher: Metrics for Evaluating Performance of Prognostic Techniques. DOI 10.1109/PHM.2008.4711436 (2008)

Characteristics of different evaluation metrics for algorithmic performance (focus on PHM) (2/3)

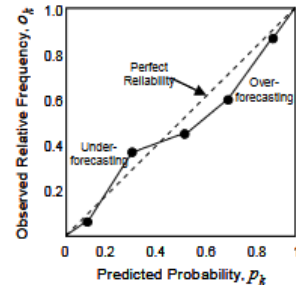
Metric Name	Definition	Description	Range
Anomaly correlation coefficient (ACC)	$ACC = \frac{\sum (\pi'(i j) - z_g(i))(z_s(i) - z_g(i))}{\sqrt{\sum (\pi'(i j) - z_g(i))^2 \sum (z_s(i) - z_g(i))^2}}$ <p>where, $z_s(i)$ is a prediction variable (e.g. $f_n^l(i)$ or $h_n^l(i)$), and $z_g(i)$ is the corresponding history data value.</p>	Measures correspondence or phase difference between prediction and observations, subtracting out the historical mean at each point. The anomaly correlation is frequently used to verify output from numerical weather prediction (NWP) models. ACC is not sensitive to error or bias, so a good anomaly correlation does not guarantee accurate predictions. In the PHM context, ACC computed over a few time-steps after t_f can be used to modify long term predictions. However, the method requires computing a baseline from history data which may be difficult to come by.	[-1 1] Perfect score = 1
Symmetric mean absolute percentage error (sMAPE)	$sMAPE(i) = \frac{1}{L} \sum_{i=1}^L \left \frac{100\Delta(i)}{(r_s^l(i) + r^l(i))/2} \right $	Averages the absolute percentage errors in the predictions of multiple UUTs at the same prediction horizon. The percentage is computed based on the mean value of the prediction and ground truth. This prevents the percentage error from being too large for the cases where the ground truth is close to zero.	[0,∞) Perfect score = 0
*Mean squared error (MSE)	$MSE(i) = \frac{1}{L} \sum_{i=1}^L \Delta(i)^2$	Averages the squared prediction error for multiple UUTs at the same prediction horizon. A derivative of MSE is Root Mean Squared Error (RMSE).	[0,∞) Perfect score = 0
Mean absolute error (MAE)	$MAE(i) = \frac{1}{L} \sum_{i=1}^L \Delta(i) $	Averages the absolute prediction error for multiple UUTs at the same prediction horizon. Using median instead of mean gives median absolute error (MdAE).	[0,∞) Perfect score = 0
Root mean squared percentage error (RMSPE)	$RMSPE(i) = \sqrt{\frac{1}{L} \sum_{i=1}^L \left \frac{100\Delta(i)}{r_s^l(i)} \right ^2}$	Square root of the average of percentage error of the prediction from multiple UUTs. A similar metric is Root median squared percentage error (RMdSPE).	[0,∞) Perfect score = 0
* these metrics can also be classified into precision based category			
Precision Based Metrics			
Sample standard deviation (S)	$S(i) = \sqrt{\frac{\sum_{i=1}^n (\Delta^l(i) - M)^2}{n-1}}$ <p>where M is the sample mean of the error</p>	Sample standard deviation measures the dispersion/spread of the error with respect to the sample mean of the error. This metric is restricted to the assumption of normal distribution of the error. It is, therefore, recommended to carry out a visual inspection of the error plots.	[0,∞) Perfect score = 0
Mean absolute deviation from the sample median (MAD)	$AD(i) = \frac{1}{n} \sum_{i=1}^n \Delta^l(i) - M $ <p>where $M = \text{median}_i(\Delta^l(i))$ and median is the $\frac{n+1}{2}$-th order statistic</p>	This is a resistant estimator of the dispersion/spread of the prediction error. It is intended to be used where there is a small number of UUTs and when the error plots do not resemble those of a normal distribution.	[0,∞) Perfect score = 0
Median absolute deviation from the sample median (MdAD)	$MAD(i) = \text{median}_i(\Delta^l(i) - M)$ <p>where $M = \text{median}_i(\Delta^l(i))$ and median is the $\frac{n+1}{2}$-th order statistic</p>	This is a resistant estimator of the dispersion/spread of the prediction error. It is intended to be used where there is a small number of UUTs and when the error plots do not resemble those of a normal distribution.	[0,∞) Perfect score = 0

Source: Saxena, Celaya, Balaban, Goebel, Saha, Saha, Schwabacher: Metrics for Evaluating Performance of Prognostic Techniques. DOI 10.1109/PHM.2008.4711436 (2008)

Characteristics of different evaluation metrics for algorithmic performance (focus on PHM) (3/3)

Robustness Based Metrics

Reliability diagram,
Brier Score



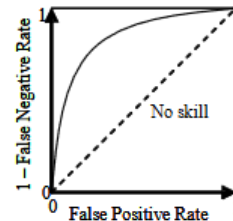
Reliability Diagram
The Brier Score computed as

$$BS = \frac{1}{K} \sum_{k=1}^K (p_k - o_k)^2$$
 is a measure of the deviation from the diagonal.

The reliability diagram plots the observed frequency against the predicted probability of a random event. In the context of prognostics, an event may be the RUL of a system lying within a given time interval, or a health feature crossing an alarm threshold within a predetermined time. The prediction of the value of RUL is not considered an event. In other words, the problem of prognostics is transformed into the classification domain. The occurrence of the event is predicted multiple times and the range of probabilities is divided into K bins like 0-5%, 5-15%, 15-25%, etc. Let us say that n_k times out of a total of N , the predicted probability falls in the probability bin k centered around p_k and out of those n_k times, the event occurs m_k times, then the corresponding observed relative frequency o_k is calculated as m_k/n_k . Reliability is indicated by the proximity of the plotted curve to the diagonal. The deviation from the diagonal gives the conditional bias. If the curve lies below the line, this indicates over-forecasting (probabilities too high); points above the line indicate under-forecasting (probabilities too low).

[0 1]
Perfect
score = 0

Receiver Operating
Characteristic
(ROC)



The area under the ROC curve can be used as a score.

ROC gives a comprehensive overview of the tradeoff between false positives and false negatives, as defined in section VIII. The ideal curve would have zero false positives and zero false negatives. Such a curve cannot realistically be achieved for real-world problems. In addition, tuning the prognostic algorithm such that a ROC can be generated may prove difficult in practice (e.g., due to lack of data or lack of tuning "parameters").

[0 1]
Perfect
score = 1

Sensitivity

$$S(i) = \frac{1}{L} \sum_{l=1}^L \left\{ \frac{\Delta M^l(i)}{\Delta_{input}} \right\}$$

Measures how sensitive a prognostic algorithm is to input changes or external disturbances. Can be assessed against any performance metric of interest. ΔM is the distance measure between two successive outputs for metric M 's value and Δ_{input} is a distance measure between two successive inputs, e.g. failure threshold, noise level, available sensor set, sampling rate, etc.

[0, ∞)
Perfect
score = 0

Discussion of relations and differences of mean absolute errors (MAE) and root mean square errors (RMSE)

- “MAE is a more natural measure of average error and (unlike RMSE) is unambiguous”
- “RMSE is an inappropriate and misinterpreted measure of average error because of 3 characteristics of a set of errors. RMSE varies with the variability within the distribution of error magnitudes and with the square root of the number of errors ($n^{1/2}$), as well as with the average-error magnitude (MAE).”
- “MBE can convey useful information, but should be interpreted cautiously since it is inconsistently related to typical-error magnitude, other than being an underestimate ($MBE \leq MAE \leq RMSE$)”

Sample data

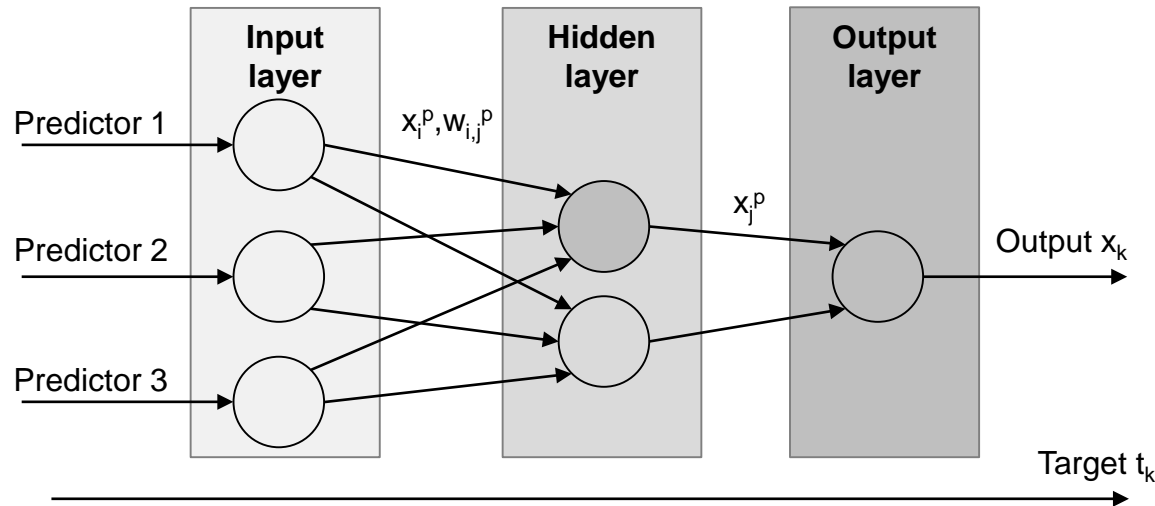
Variable	Case 1	Case 2	Case 3	Case 4	Case 5
e_1	2	1	1	0	0
e_2	2	1	1	0	0
e_3	2	3	1	1	0
e_4	2	3	5	7	8
$\sum e_i $	8	8	8	8	8
MAE	2	2	2	2	2
$\sum e_i ^2$	16	20	28	50	64
RMSE	2.0	2.2	2.6	3.5	4.0

Source: Willmott, Matsuura: Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. In: J. of Clim Res, Vol. 30: 79–82 (2005)

DIAGNOSES MODELS FOR AIRCRAFT FUEL FLOW

Own studies for aircraft fuel flow calculations identified two promising machine learning methods.

Artificial Neural Network / Multilayer Perceptrons



Perceptron transfer function

$$x_j^p = f\left(\sum_{i=1}^n w_{j,i}^p \cdot x_i^p\right)$$

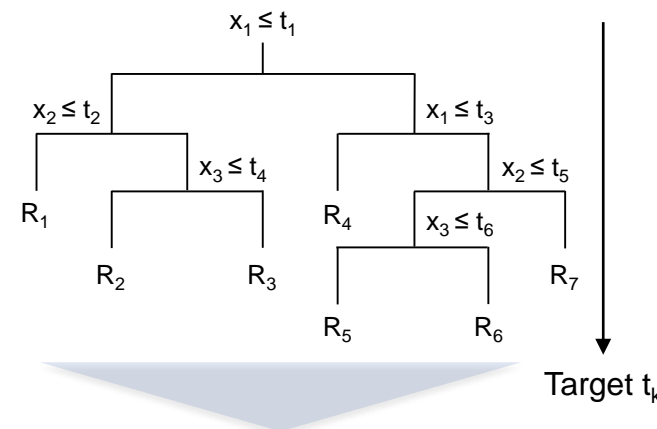
Perceptron activation function

$$a^p := \begin{cases} 0 & \text{falls } \sum_{j=1}^n w_{j,i}^p \cdot x_j^p < \theta \\ 1 & \text{sonst} \end{cases}$$

Quality criterion: MSE

$$J = \min \left(E(w_{j,i}^p) = \frac{1}{2} \sum_{j=1}^n (t_k - x_k)^2 \right)$$

Decision & Model Trees / Ensemble Method (Bagging)



Entropy

$$E(P) = \sum_{i=1}^n P_i \cdot \log_2(P_i)$$

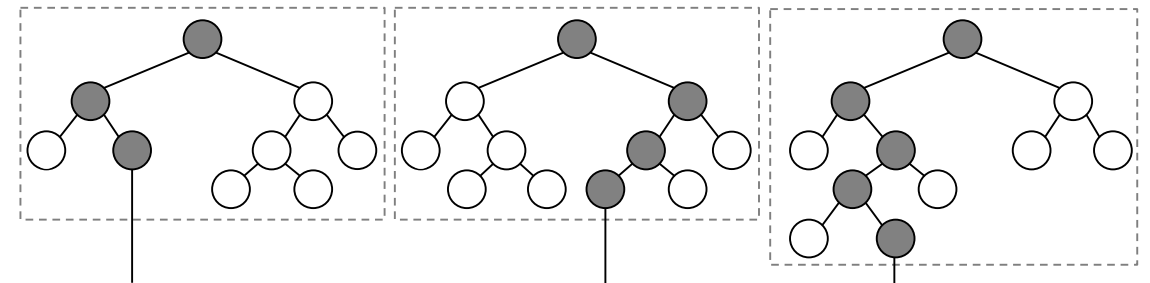
Informativeness

$$I(S) := 1 - E(P)$$

Gain of information

$$G(S, A) = E(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} E(S_i)$$

Random Forest (bootstrapped aggregated)



Voting (classification) / averaging (regression)

Neural networks require specifications for determining the “operation” mode.

- **Topology settings** such as number of hidden layers and perceptrons
- **Optimization algorithms** for the gradient method
- **Weight initialization methods:** random choice of initial weights from a normal distribution, standard deviation dependent on the number of neurons
- **Perceptron activation function:** in the past: sigmoid / tanh; nowadays: rectified linear units

Perceptron count

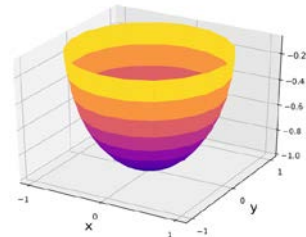
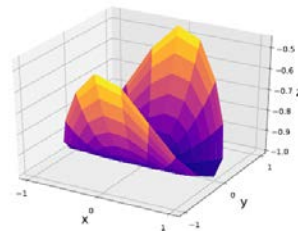
$$n = \sqrt{n_{\text{Input}} \cdot n_{\text{Output}}}$$
$$n = n_{\text{Input}} \cdot \log(2)$$

Weight initialization

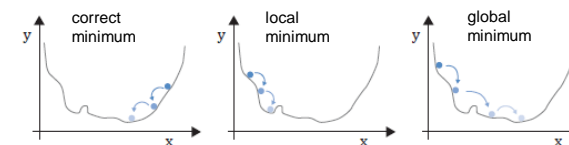
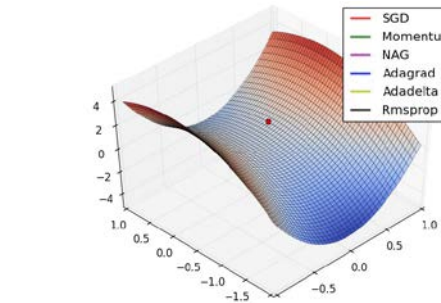
Truncated normal : $\sigma = \frac{1}{\sqrt{n_{\text{in}}}}$

He normal: $\sigma = \frac{2}{\sqrt{n_{\text{in}}}}$

Training problem: gradient descent Example: Loss function shapes

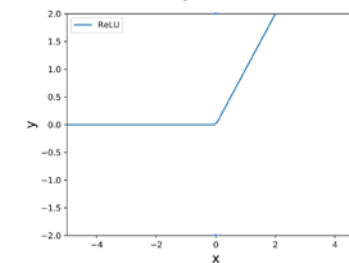


Optimization performance problem

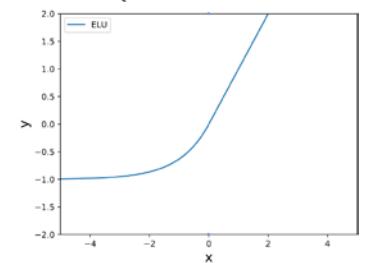


Activation functions ReLU ELU

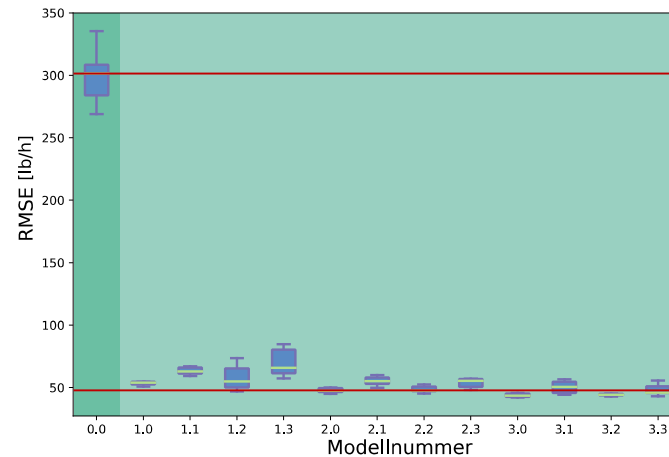
$$f(x) = \begin{cases} x & \text{für } x > 0 \\ 0 & \text{für } x \leq 0 \end{cases}$$



$$f(x) = \begin{cases} x & \text{für } x > 0 \\ \alpha(e^x - 1) & \text{für } x \leq 0 \end{cases}$$



Performance comparison of neural network configurations regarding the aircraft fuel flow calculation use case.

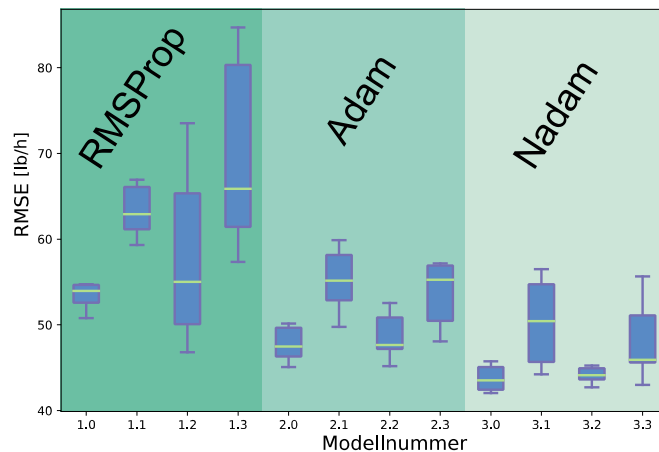


not optimized

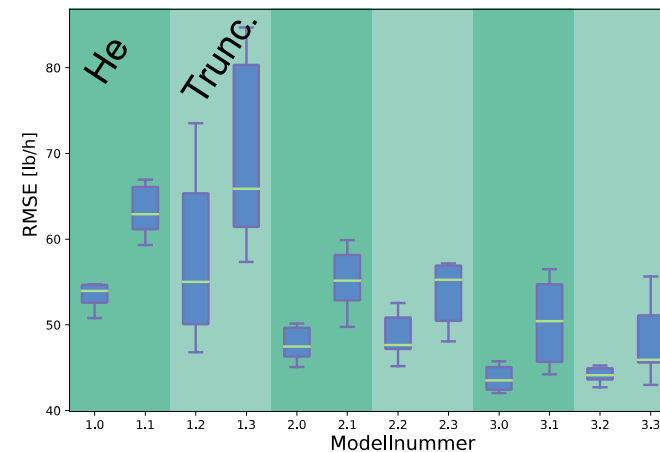
- mini batch gradient descent
- sigmoid

„Matlab“ benchmark: Bayesian regularization

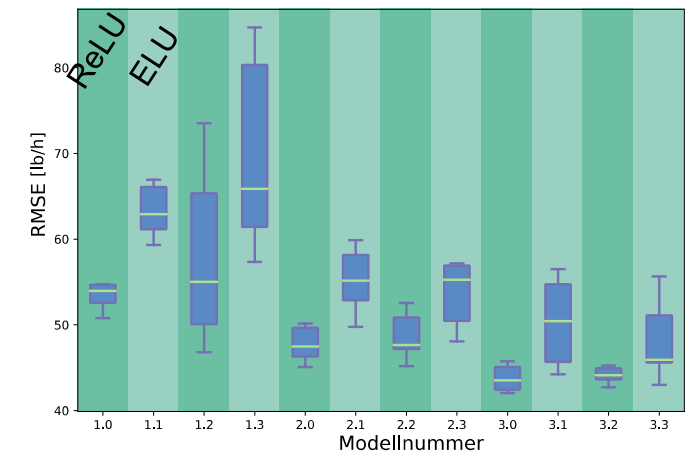
Optimization algorithms



Weight initialization methods



Activation functions



MACHINE LEARNING RESULTS FOR AIRCRAFT FUEL FLOW CALCULATIONS

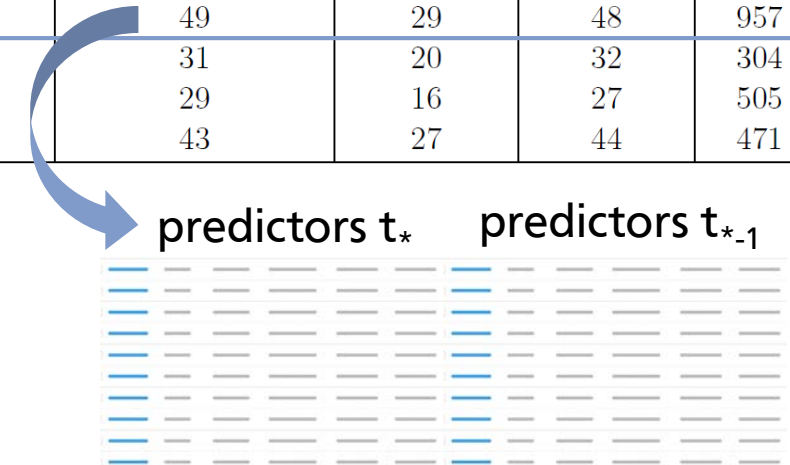
First steps of modelling are broadly diversified, then promising ones are evaluated, selected and optimized.

Comparison of different models

Evaluation type	MAE	RMSE	Training time, sec
MLP	32	51	2165
M5Rules	37	65	289
M5P	37	68	57
RandomForest	57	91	53
REPTree	91	149	10
RandomTree	109	196	1
LinearRegression	130	186	26
DecisionTable	180	294	174

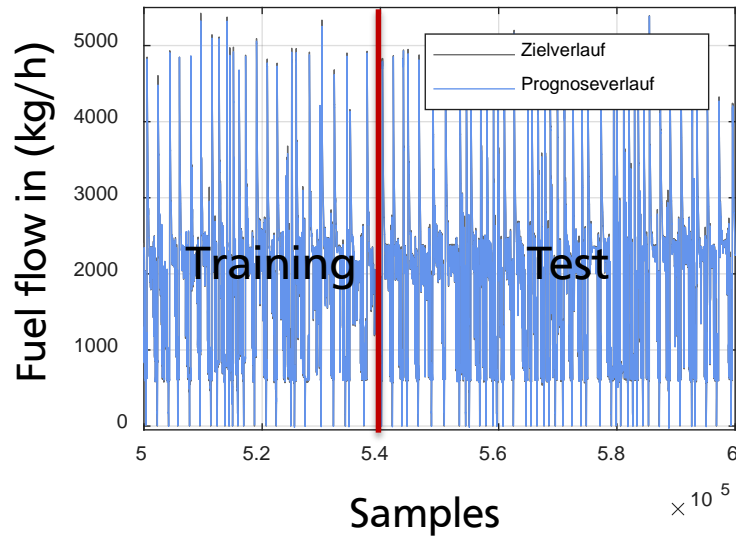
Comparison of different use cases and feature sets

Dataset	MAE $S_{Train, CV_{x10}}$	MSE $S_{Train, CV_{x10}}$	MAE S_{test}	MSE S_{test}	Time, s
PH_{all}	32	58	32	63	361
$A_{Doubled}$	29	49	29	48	957
PH_4	20	31	20	32	304
PH_5	17	29	16	27	505
PH_6	27	43	27	44	471

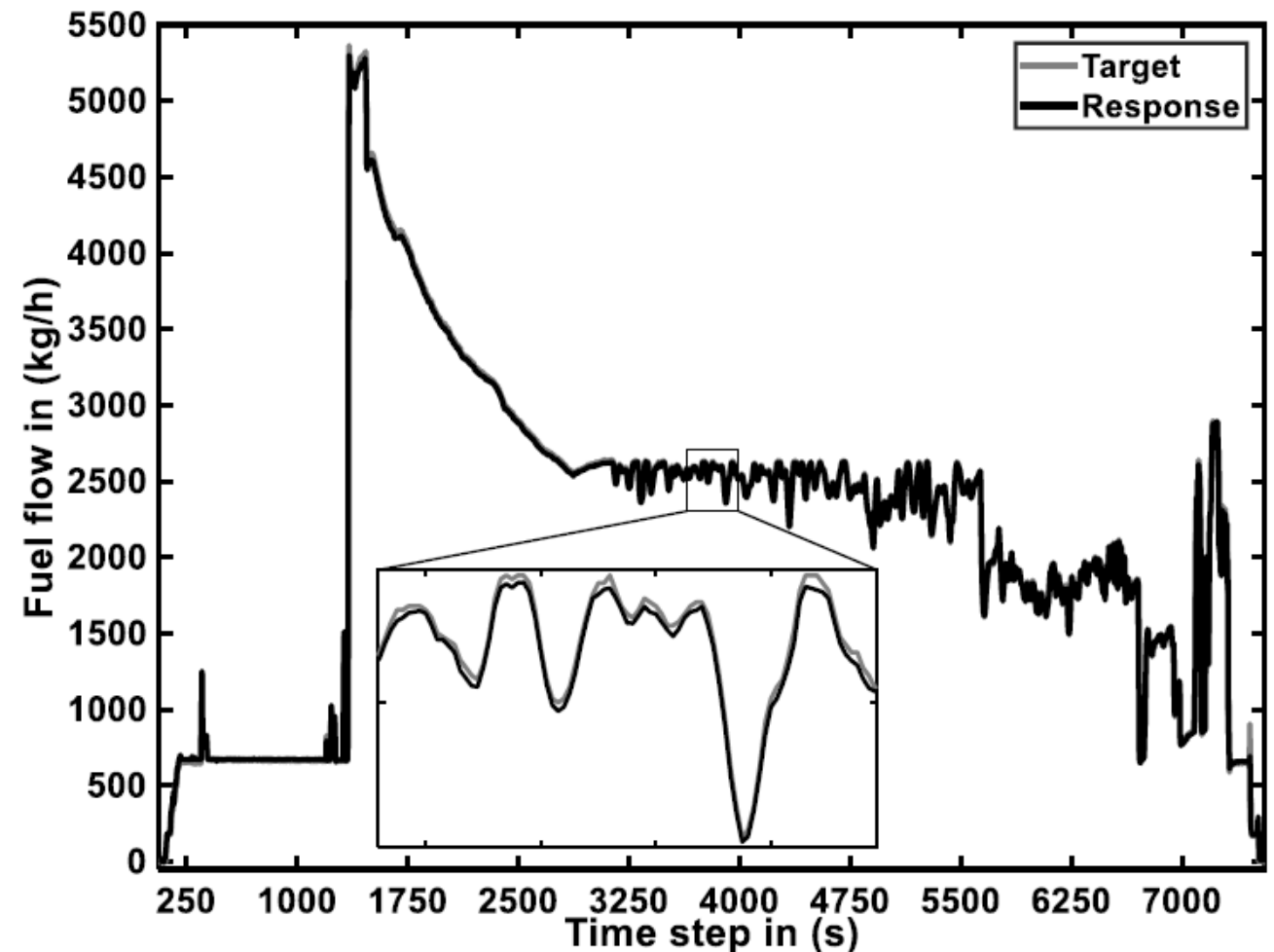


- **Computing time:** ANN can have a 10 times higher computing time than random forests
- **Feature generation (predictor doubling):** The calculation time increases nonlinearly with the number of predictors used.
- **Decision trees:** Model trees like M5 and Random Forest have less errors than regression trees.
- **Normalization and standardization:** no significant differences can be observed in the use of decision tree procedures
- **Shuffling:** Mixing the instances can improve errors by a factor of two in the use of ANN

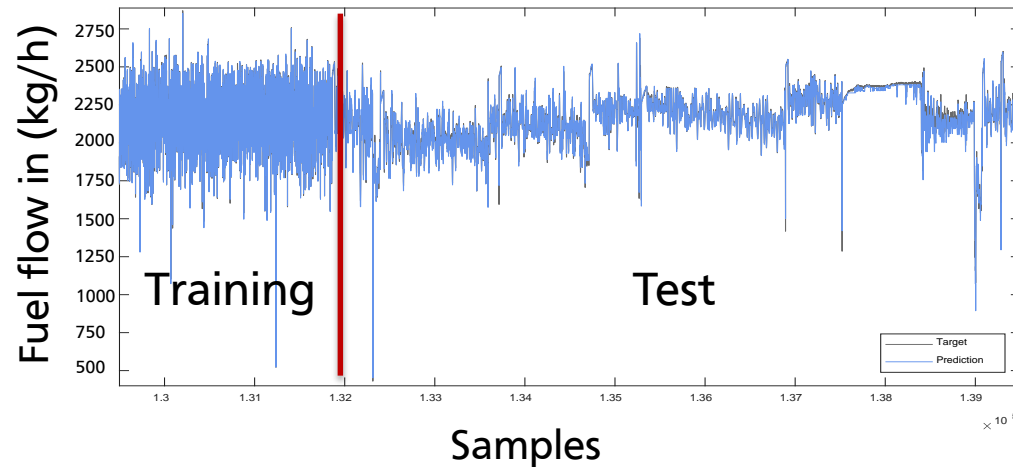
Neural networks calculate the fuel flow for flight missions with errors less than 1%.




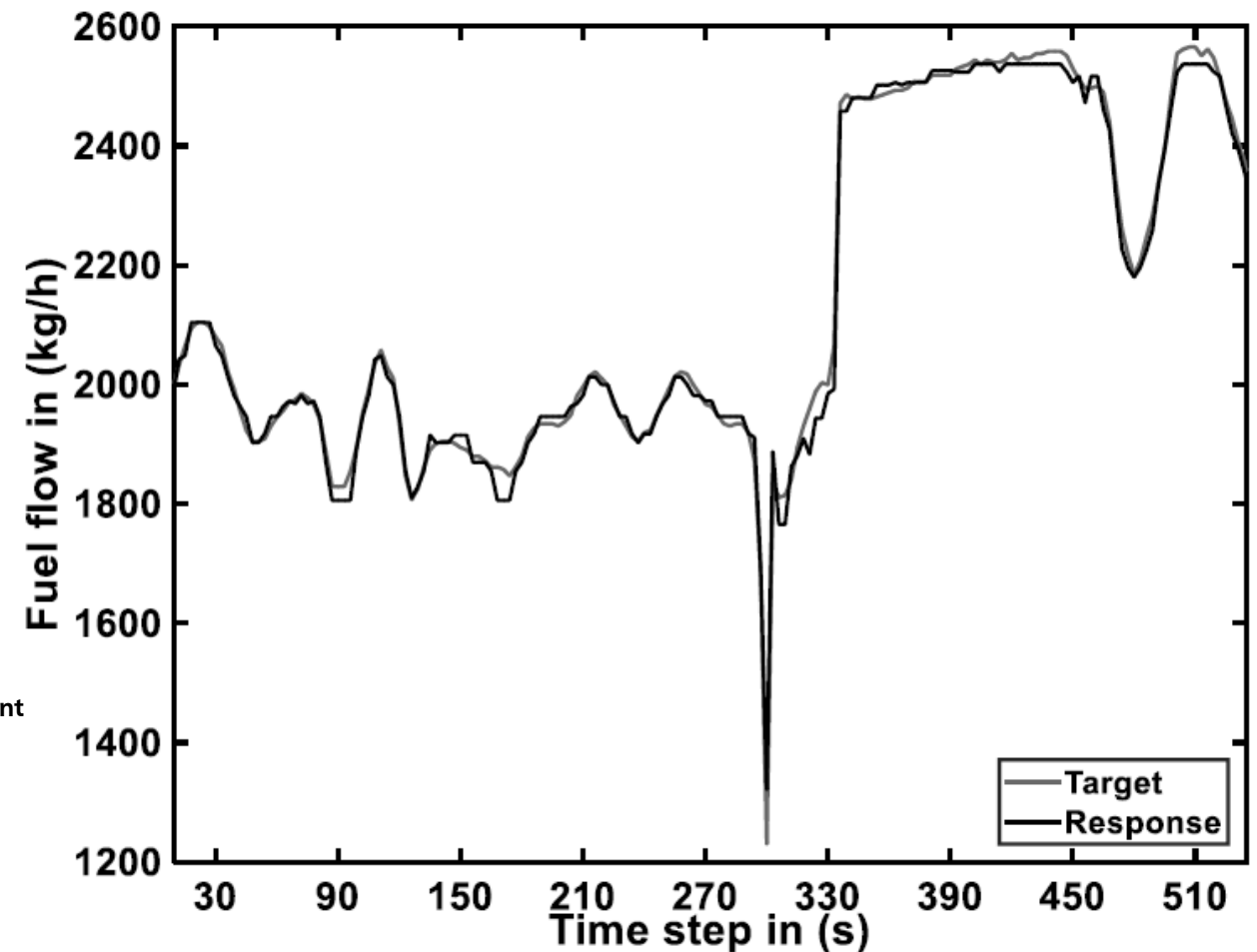
Neural network	Training	Test
MAF MRE	7,25 kg/h 0,4 %	15,88 kg/h 0,8 %
MQF	12,70 kg/h	22,23 kg/h
R ²	0,9998	0,9995
Random Forest	Training	Test
MAF MRE	24,04 kg/h 1,3 %	39,46 kg/h 2,1 %
MQF	36,74 kg/h	66,67 kg/h
R ²	0,9987	0,9952



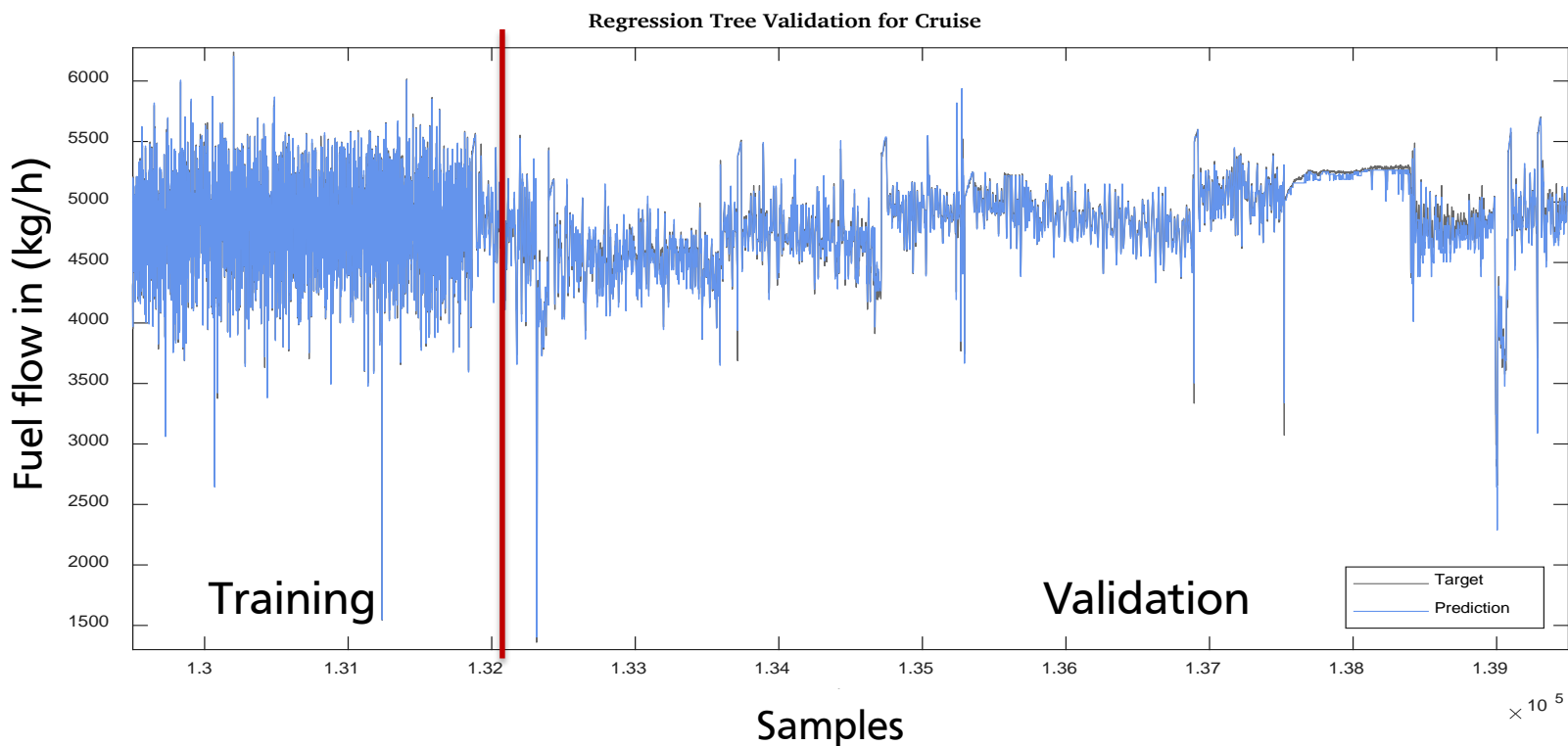
For the cruise phase, however, decision tree procedures perform better.



Neural network	Training	Test	 Risk of over-adjustment
MAE MRE	4,5 kg/h 0,13 %	41,3 kg/h 1,90 %	
RMSE	4,08 kg/h	68,95 kg/h	
R ²	0,995	0,880	
Random Forest	Training	Test	
MAE MRE	8,40 kg/h 0,74 %	27,30 kg/h 1,26 %	
RMSE	14,30 kg/h	36,74 kg/h	
R ²	0,998	0,963	



Feature reduction: Using a quarter of the predictors creates comparable results, but computing time is divided by three.



Predictor Cluster	DESCRIPTION
ALT	Pressure altitude
ALTR, IVV	Altitude rate, inertial vertical velocity
BLAC / VRTG	Body longitudinal / vertical acceleration
CAS, GS	Calculated air speed, ground speed
DA	Drift Angle
EGT 1-4	Exhaust Gas Temperatures 1-4
FPAC	Flight Path Acceleration
FQTY 1-4 / Weight	Fuel quantity start / fuel weight
LATG / LONG	Latitude / longitude acceleration
LATP / LONP	Latitude / longitude position
MACH	Mach
MH	Magnetic heading
N1 1-4	N1 shaft speed
PS	Pressure (static (mean), total)
SAT / TAT	Static / total air temperature
TAS	True air speed
TH	True heading
WD / WS	Wind direction / wind speed

36 Features	Training Data	Validation Data
MAE / MRE	13 kg/h / 0.26 %	58 kg/h / 1.2 %
RMSE	18 kg/h	81 kg/h
R ² (NSE)	0.998	0.963

BUSINESS CASE BENCHMARK WITH A PHYSICAL MODEL

Aircraft performance model of the Base of Aircraft Data (BADA) from EUROCONTROL

- Total energy model

$$(T - D)V_{TAS} = mg \frac{dh}{dt} + mVTAS \frac{dV_{TAS}}{dt}$$

- BADA fuel consumption model

$$FF_{nom} = f_0 + \left(f_1 + f_2 V_{TAS} - f_3 V_{TAS}^2 \right) T$$

$$FF_{min} = f_4 - f_5 h$$



Nominal fuel-flow rate model coefficients

Engine Type	f_0	f_1	f_2	f_3
Jet	0	$\left(\frac{1}{6 \times 10^4} \right) C_{f1}$	$\left(\frac{1}{6 \times 10^4} \right) \left(\frac{C_{f1}}{C_{f2}} \right)$	0
Turboprop	0	0	$\left(\frac{1}{6 \times 10^7} \right) C_{f1}$	$\left(\frac{1}{6 \times 10^7} \right) \left(\frac{C_{f1}}{C_{f2}} \right)$
Piston	$\left(\frac{1}{60} \right) C_{f1}$	0	0	0

Minimum fuel-flow rate model coefficients

Engine Type	f_4	f_5
Jet	$\left(\frac{1}{60} \right) C_{f3}$	$\left(\frac{1}{60} \right) \left(\frac{C_{f3}}{C_{f4}} \right)$
Turboprop	$\left(\frac{1}{60} \right) C_{f3}$	$\left(\frac{1}{60} \right) \left(\frac{C_{f3}}{C_{f4}} \right)$
Piston	$\left(\frac{1}{60} \right) C_{f3}$	0

“BADA 3 model demonstrates the ability to predict aircraft performances with a mean root mean square (RMS) error in vertical speed lower than 100 fpm and a fuel flow error less than 5%” (at nominal operating conditions and temperature deviations (ΔT) from the International Standard Atmosphere (ISA) conditions ranging from ISA+0 to ISA+20)).

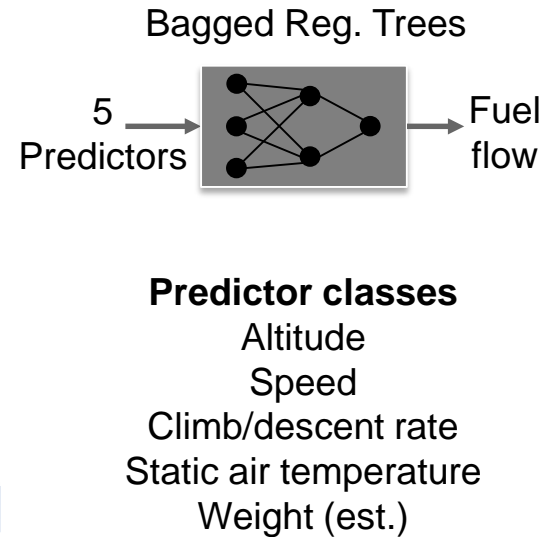
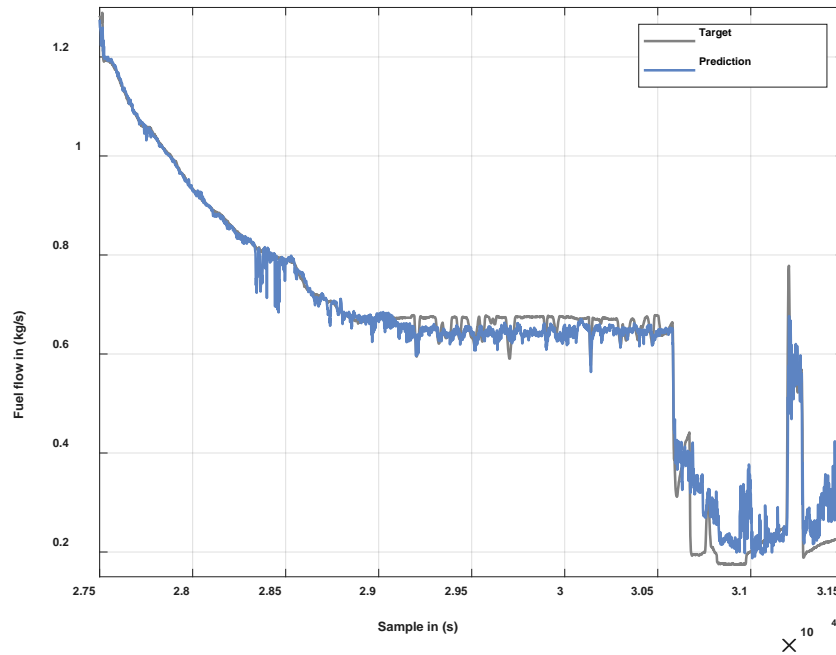
Excerpt of the aircraft specific calculation factors

Category	Parameter and Description	Category	Parameter and Description
aircraft type	n_{eng} – number of engines [-] engine type – Jet/Turboprop/Piston wake category – Heavy/Medium/Light	engine thrust	$C_{Tc,1}$ – 1 st max. climb thrust coefficient [N] $C_{Tc,2}$ – 2 nd max. climb thrust coefficient [ft] $C_{Tc,3}$ – 3 rd max. climb thrust coefficient [1/ft ²] $C_{Tc,4}$ – 1 st thrust temperature coefficient [°C] $C_{Tc,5}$ – 2 nd thrust temperature coefficient [1/°C] $C_{Tdes,low}$ – low alt. descent thrust coefficient [-] $C_{Tdes,high}$ – high altitude descent thrust coef. [-] h_{des} – transition altitude [ft] $C_{Tdes,app}$ – approach thrust coefficient [-] $C_{Tdes,ld}$ – landing thrust coefficient [-] $V_{des,ref}$ – reference descent speed [kt] $M_{des,ref}$ – reference descent Mach number [-]
mass	m_{ref} – reference mass [t] m_{min} – minimum mass [t] m_{max} – maximum mass [t] m_{pyld} – maximum payload [t]		fuel flow
flight envelope	V_{MO} – max. operating speed [kt] M_{MO} – max. operating Mach number [-] h_{MO} – max. operating altitude [ft] h_{max} – max. altitude at MTOW and ISA [ft] G_W – weight gradient on max. altitude [ft/kg] G_t – temp. gradient on max. altitude [ft/C]		
aero-dynamics	S – reference wing surface area [m ²] $C_{D0,CR}$ – parasitic drag coefficient (cruise) [-] $C_{D2,CR}$ – induced drag coefficient (cruise) [-] $C_{D0,AP}$ – parasitic drag coefficient (approach) [-] $C_{D2,AP}$ – induced drag coefficient (approach) [-] $C_{D0,LD}$ – parasitic drag coefficient (landing) [-] $C_{D2,LD}$ – induced drag coefficient (landing) [-] $C_{D0,LDG}$ – parasitic drag coef. (landing gear) [-] C_{M16} – Mach drag coefficient [-] $(V_{stall})_i$ – stall speeds for TO,IC,CR,AP,LD [kt] $C_{LBO(M-Q)}$ – Buffet onset lift coef. [-] *jets only* K – Buffeting gradient [1/M] *jets only*	ground operation	TOL – take-off length [m] LDL – landing length [m] span – wingspan [m] length – aircraft length [m]
		<i>Note: Units shown are valid for jet aircraft only; Some units may vary for turboprop and piston aircraft.</i>	

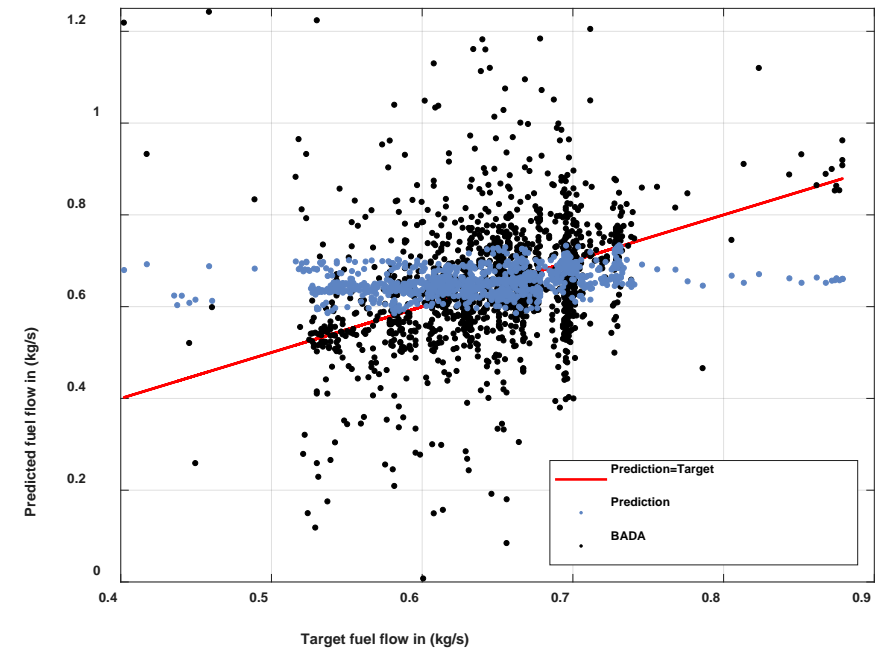
Source: https://depositonce.tu-berlin.de/bitstream/11303/1773/1/Dokument_33.pdf

The processing of less information is bought with an increased error, but still outperform conventional models.

Flight mission progression



Target response plots for data model & BADA

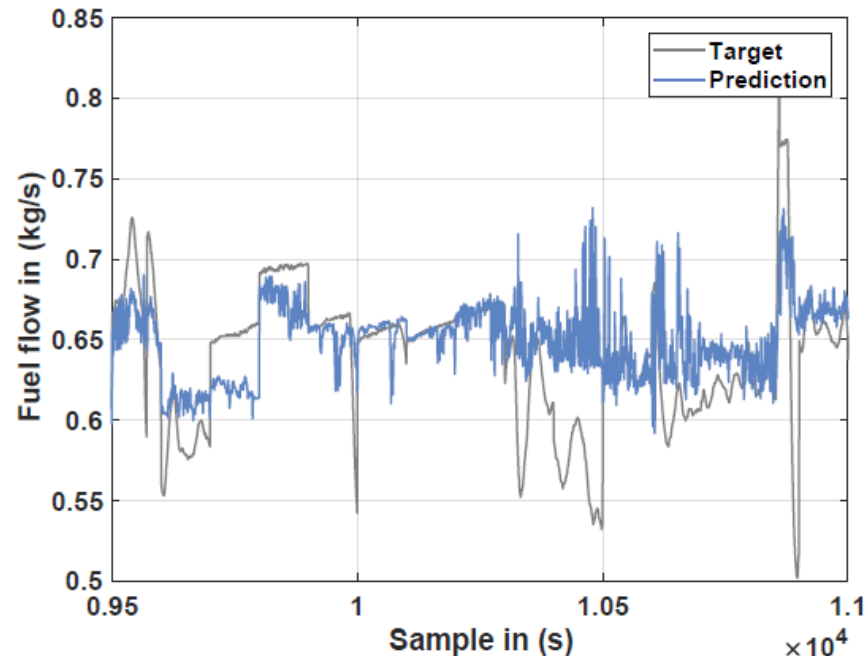


Flight missions	Training	Test
MAE in (kg/s)	.007	.029
MRE in (%)	1.1	4.4
RMSE in (kg/s)	.01	.04
R ² in (%)	94.2	48.6

Cruise	Training	Test
MAE in (kg/s)	.014	.036
MRE in (%)	2.3	5.7
RMSE in (kg/s)	.025	.053
R ² in (%)	98.9	94.7

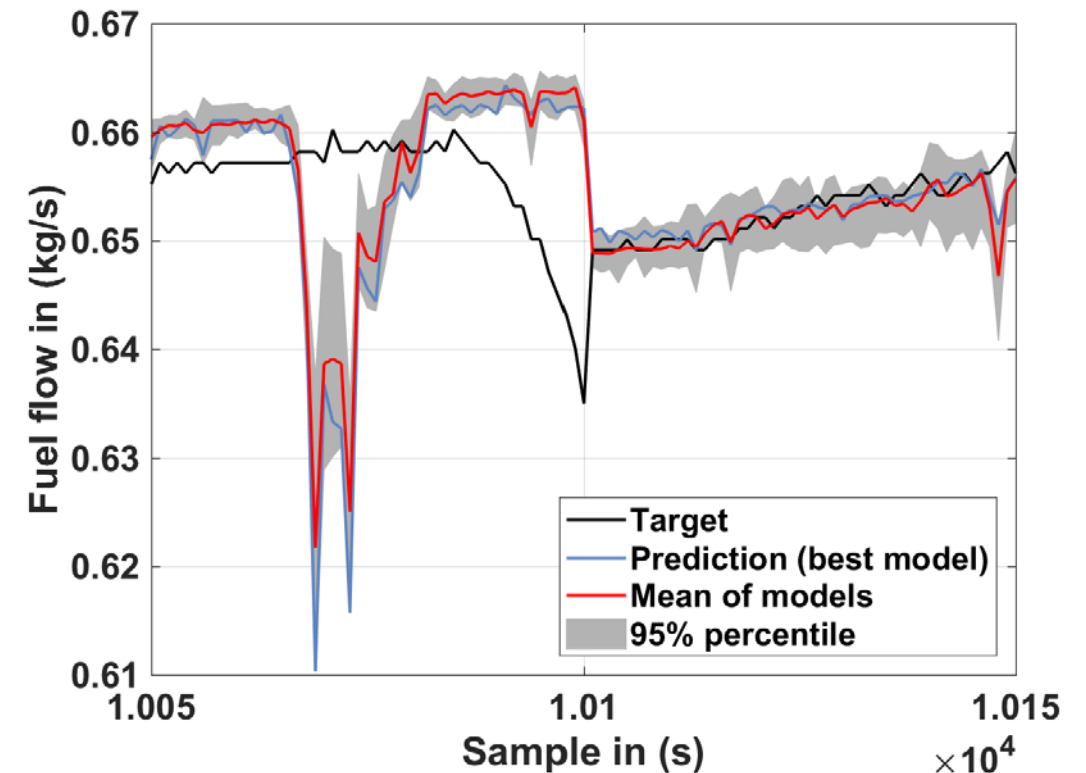
Greater robustness of the model and good generalizability should be preferred to high accuracy.

Cruise consumption



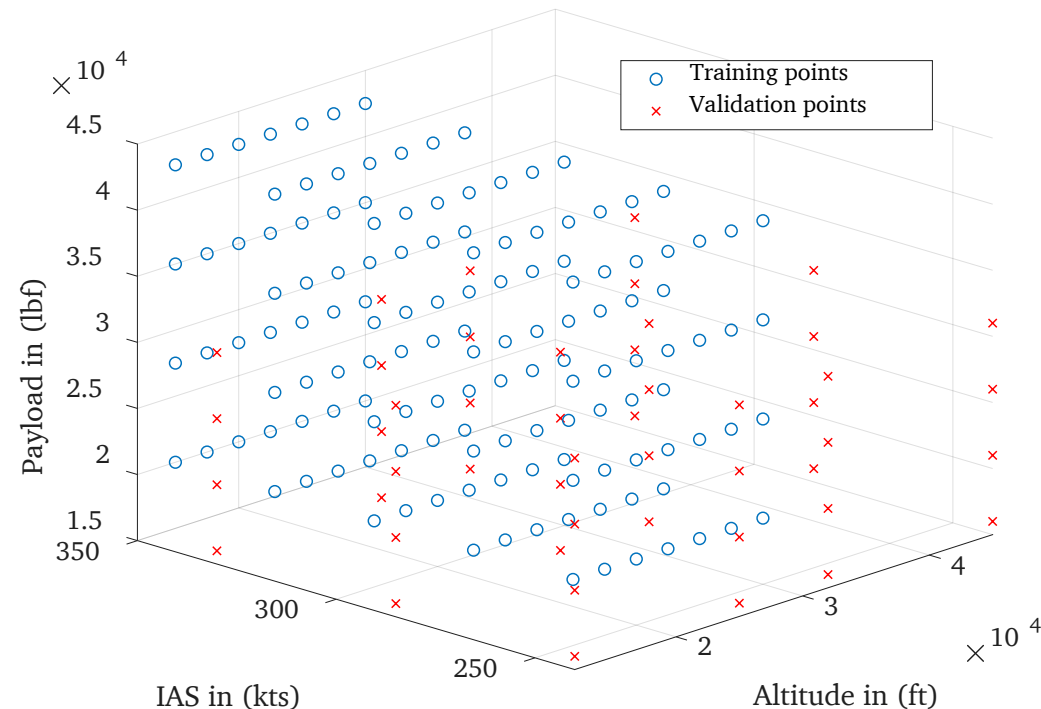
	Training	Test
<i>MAE</i> in (kg/s) <i>MRE</i> in (%)	0.014 2.3	0.036 5.7
<i>RMSE</i> in (kg/s)	0.025	0.053
<i>R</i> ² in (%)	98.9	94.7

Scattering of different models (n=100)



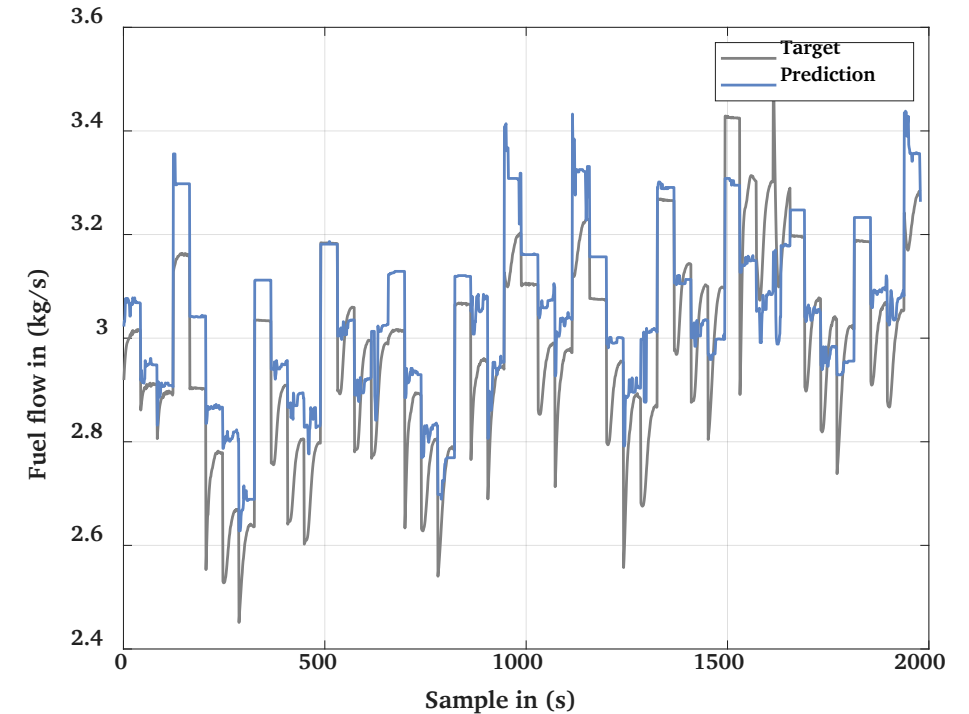
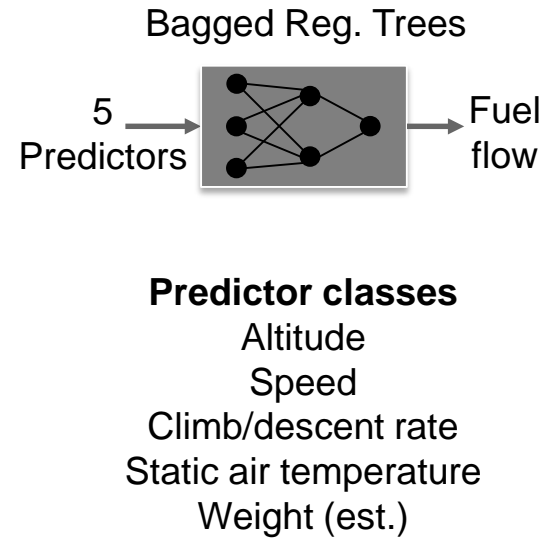
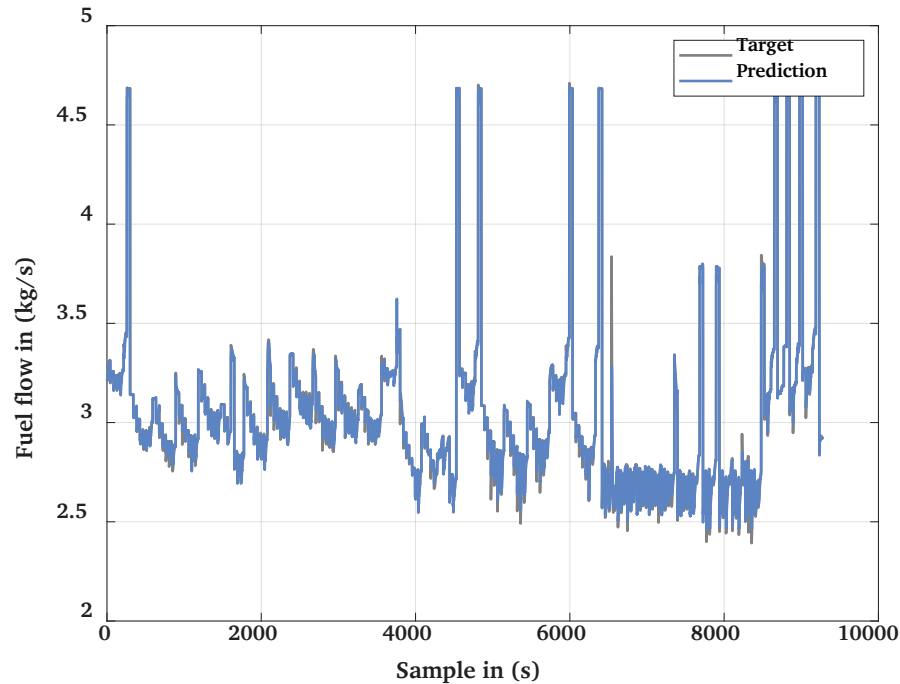
IMPACT OF INSUFFICIENT DATA QUALITY

Aircraft dependent static map points generate a database by a full factorial design of experiments.



Training data validation data	Interval	Step size
Altitude in (1000 ft)	[15, 30] [16, 31]	2,5 5
Indicated airspeed in (kts)	[250, 350] [240, 330]	25 45
Payload in (1000 lbf)	[20, 42.5] [12, 45]	7,5 {13, 7, 13}

Due to a too low information content of the database, the model cannot extrapolate sufficiently (underfitting).



	Training	Test
MAE in (kg/s) MRE in (%)	0.01 0.35	0.09 3.06
$RMSE$ in (kg/s)	0.02	0.11
R^2 in (%)	99.75	65.49

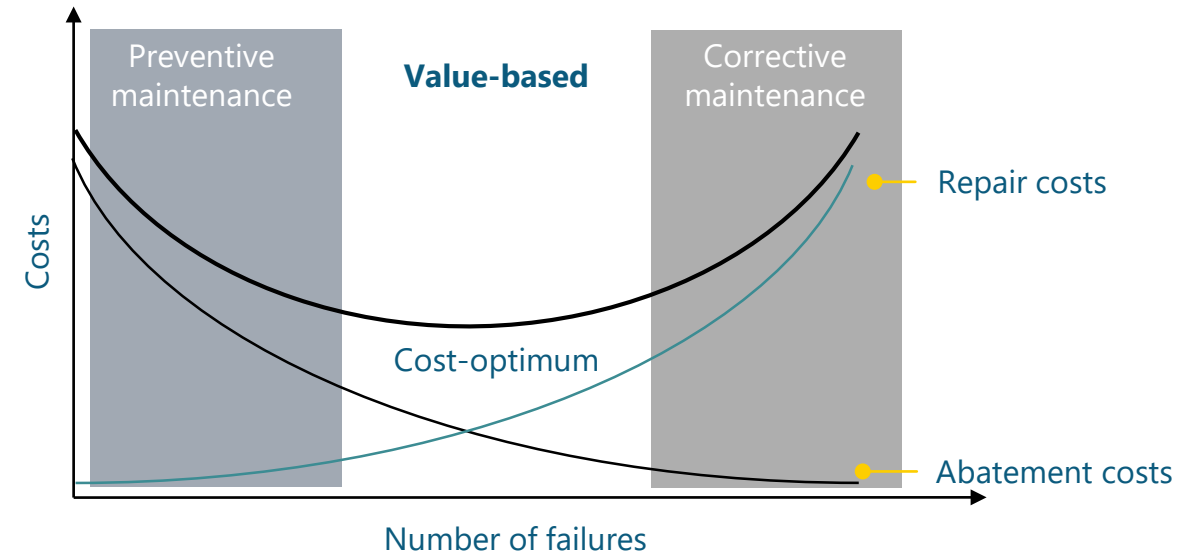
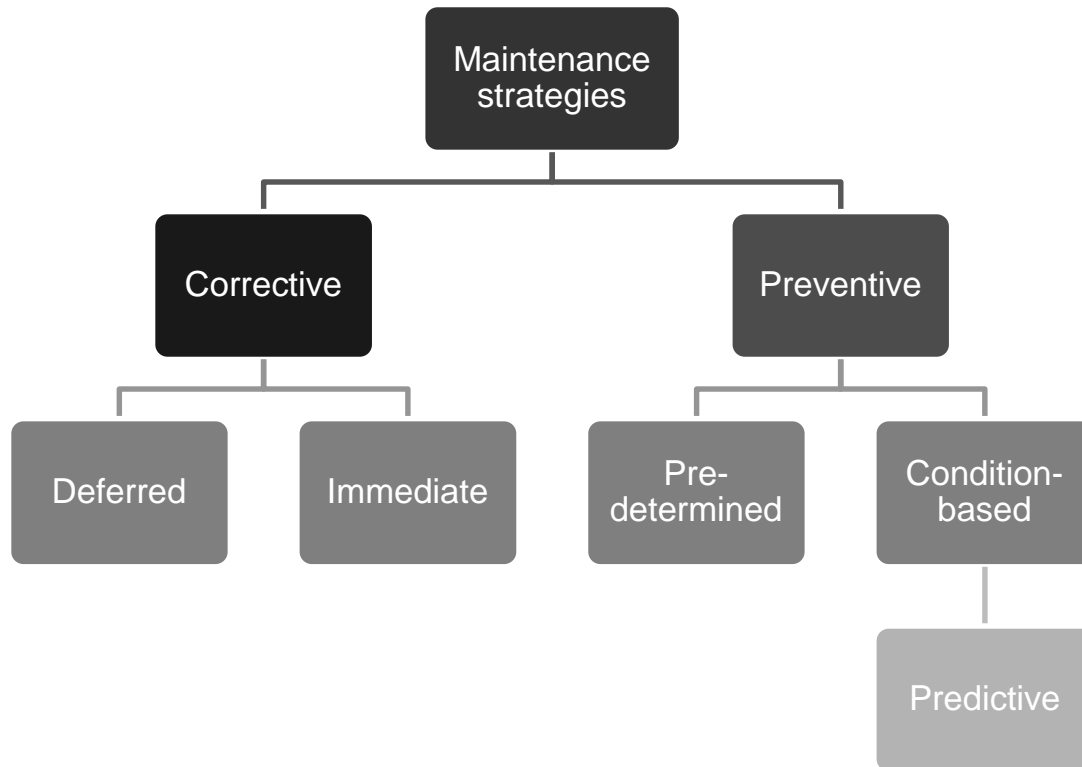
 **Risk of Under-adjustment**

Health assessment and component diagnosis

DIAGNOSIS IN PHM CONTEXT

What is PHM?

PHM: Prognostics and Health Management

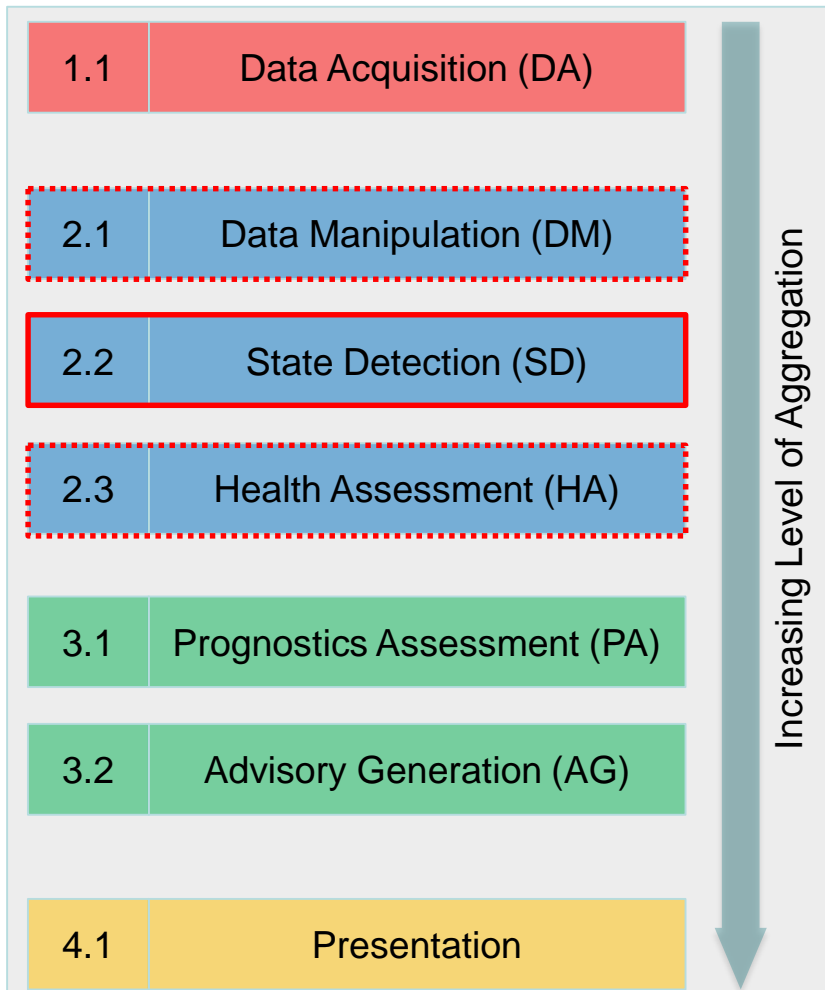


➤ **PHM is an engineering approach to find the optimum:**

- It enables health assessment
- Prediction of future state
- Improve knowledge on the system

Diagnosis is the beginning in OSA-CBM

Open System Architecture for Condition Based Maintenance



- Diagnosis focusses on part 2 in OSA-CBM
- Given dataset is manipulated in order to find or create features for state detection
- Health assessment by comparing the state with run to failure data, threshold values or similar
- The Health indicator can then be used for prognosis of the remaining useful lifetime (RUL)

General approach

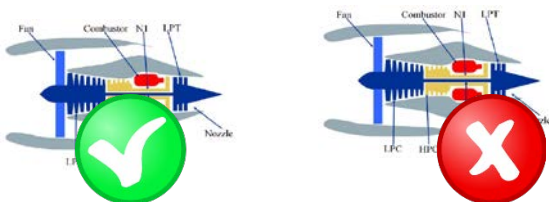
Different ways are possible to face the topic

Health assessment

Classification

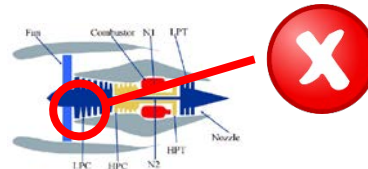
Binary

Healthy / not healthy



Multiclass

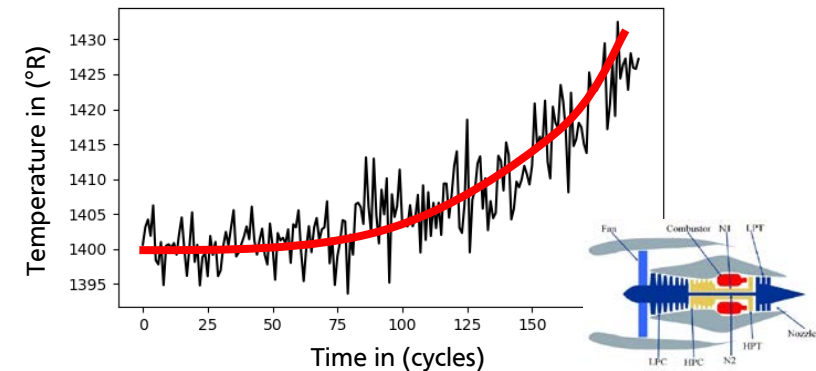
Fault 1, Fault 2, healthy, ...



Source: Saxena PHM 2008

Regression

Continuous supervising the health
Indicator (degradation curve)



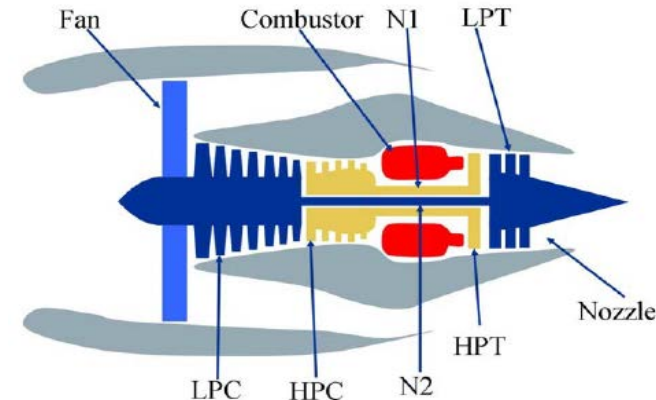
Example: Statistical Data Understanding

Dataset provided by NASA (CMAPSS – turbofan simulation)

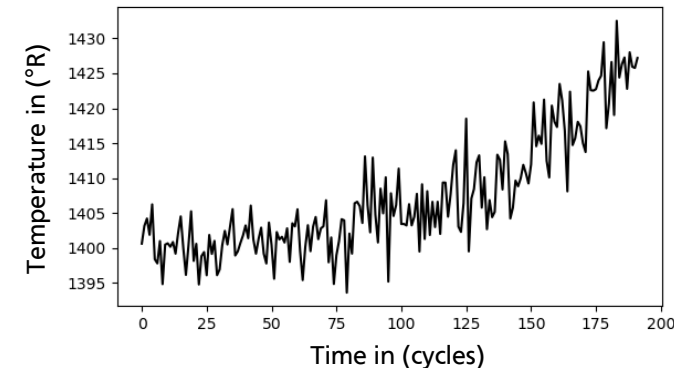
- Task:
 - Diagnosis of health index (HI)
 - Prognosis of the rest of useful lifetime (RUL)

Variables:

- 3 operational settings
 - 21 sensor variables
 - Artificial noise overlain
- Some features/targets are not directly measurable -> HI indicator
- data processing to eliminate or at least attenuate unwanted signal components



Source: Saxena PHM 2008



Dataset available for free under:

<https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>

How to get to the health state

Classification (supervised learning problem)

Binary classification (healthy, not healthy)

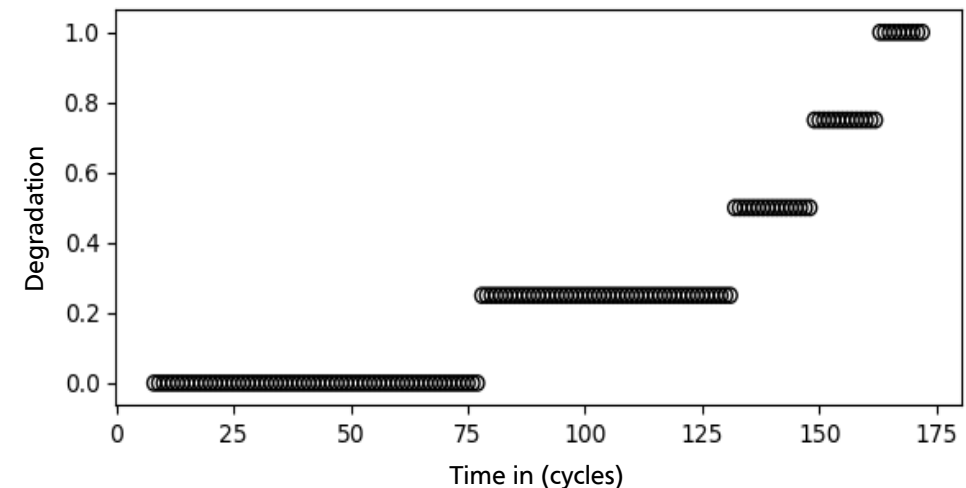
Requirements:

- Labels for health state
- Time series data as input

Multiclass classification

Requirements (same as binary):

- Different fault information or
- Higher resolution of the health index



Creating a health indicator

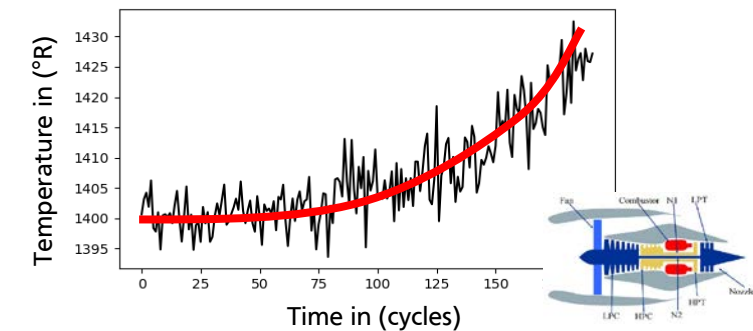
Regression (unlabeled data, anomaly detection)

Different approaches possible:

1. Creating a health indicator from given measurement variables

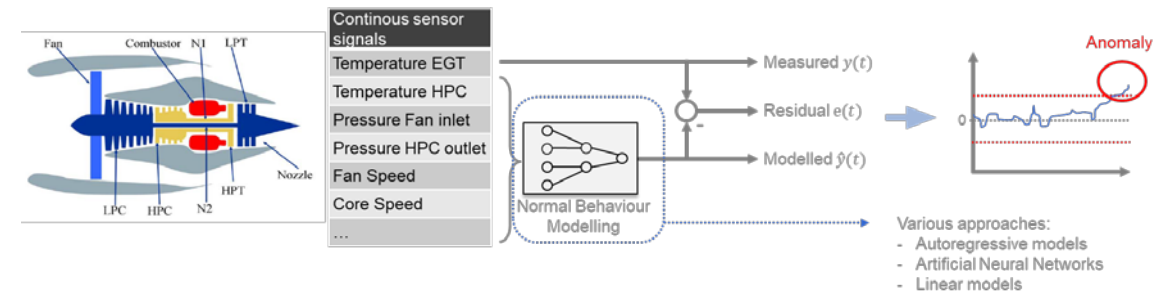
→ Degradation curve

$$HI = e^{\lambda t}$$



2. Normal Behaviour modelling

→ trend monitoring tool,
anomaly detection



Creating a health indicator

Synthetically by feature engineering

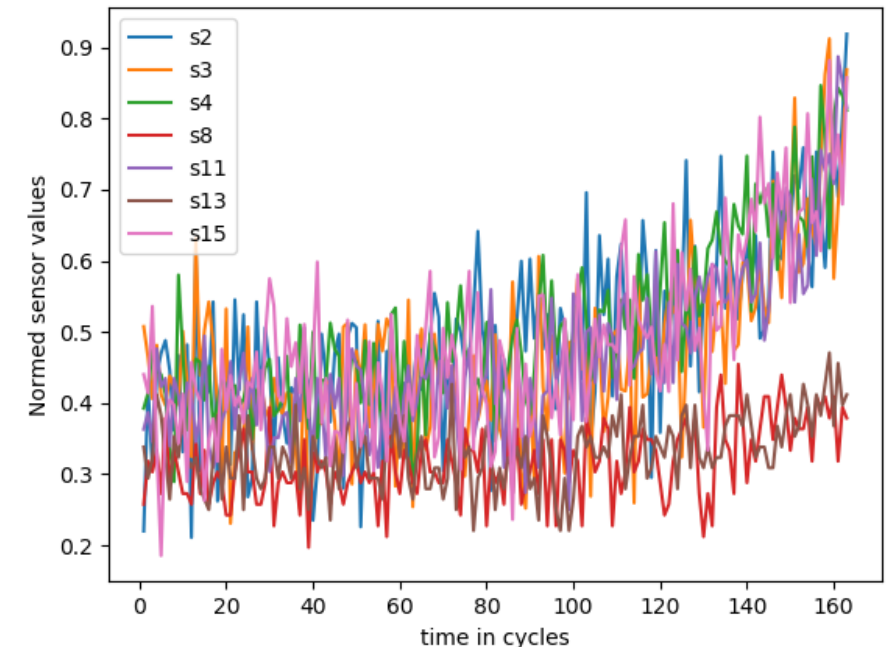
Different approaches possible:

1. Creating a health indicator from given measurement variables

→ Degradation curve

Exemplary tools:

- Feature selection
 - Choose the most important feature
 - Merge distinct features and normalize them
 - Feature extraction
 - Taking the first PC of the PCA
- the 1st PC explains about 60 % of the variance in this case



Creating a health indicator

Synthetically by feature engineering

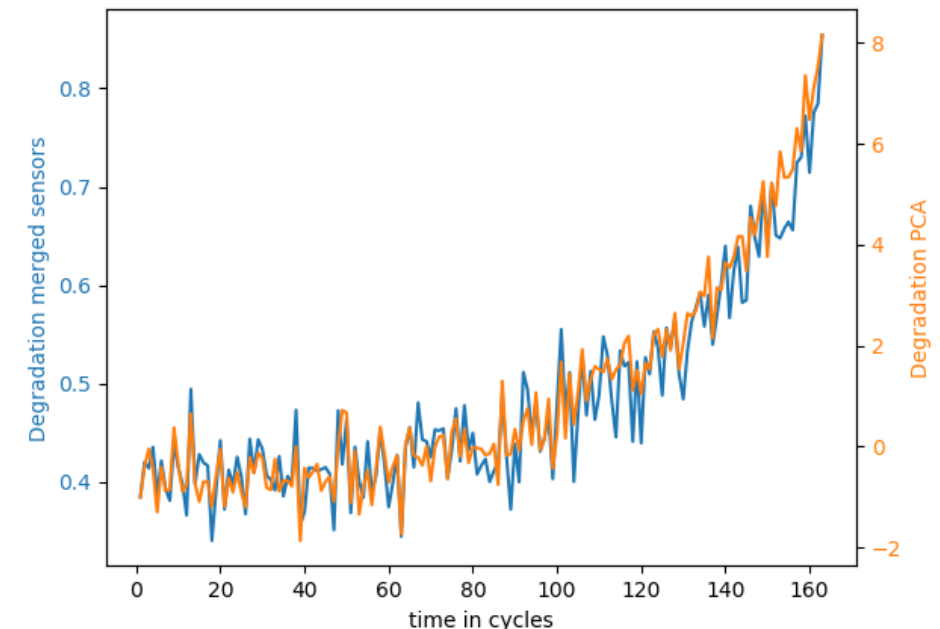
Different approaches possible:

1. Creating a health indicator from given measurement variables

→ Degradation curve

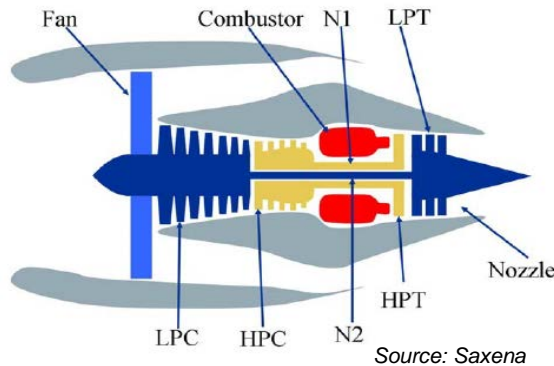
Exemplary tools:

- Feature selection
 - Choose the most important feature
 - Merge distinct features and normalize them
- Feature extraction
 - Taking the first PC of the PCA
the 1st PC explains about 60 % of the variance in this case

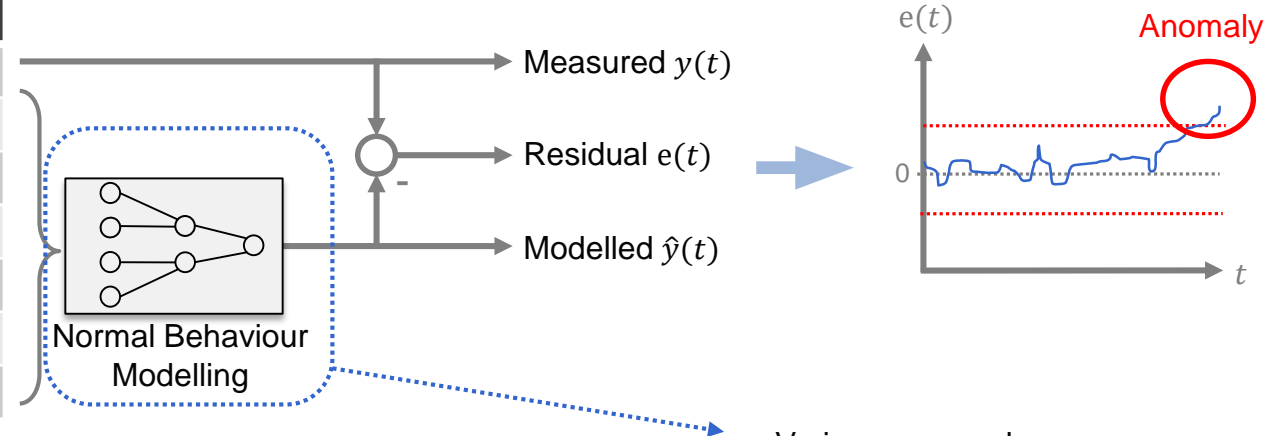


Creating a health indicator

By normal behaviour modelling (regression)



Continuous sensor signals
Temperature EGT
Temperature HPC
Pressure Fan inlet
Pressure HPC outlet
Fan Speed
Core Speed
...



Various approaches:

- Autoregressive models
- Artificial Neural Networks
- Linear models
- Decision Trees
- ...

- Modelling of important signals through the remaining
- Broad range of regression methods possible
- Comparison of the modelled and actually measured signals
→ plotting of the residuals
- Definition of thresholds or „intelligent“ outlier methods
- If there are many signals to be monitored, use of classification methods might be feasible for fault diagnosis

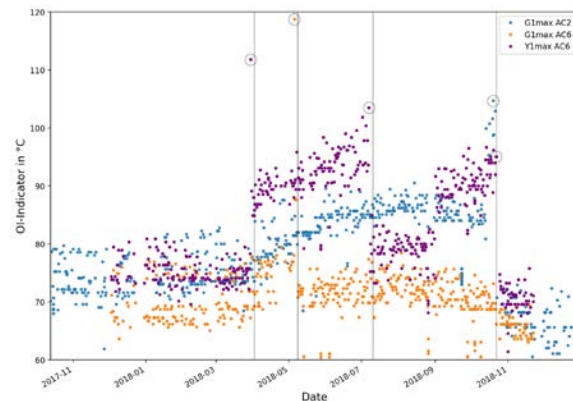
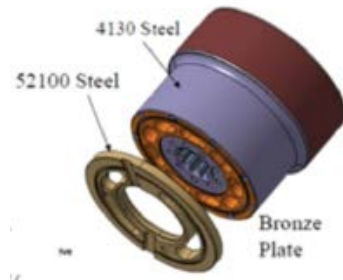
➤ **Possible approach for Hackathon Group 3 & 4 !**

Source: Weinert, J. and Watson, S.J. *Wind Turbine Fault Detection by Normal Behaviour Modelling*
<https://dSPACE.lboro.ac.uk/2134/22532>

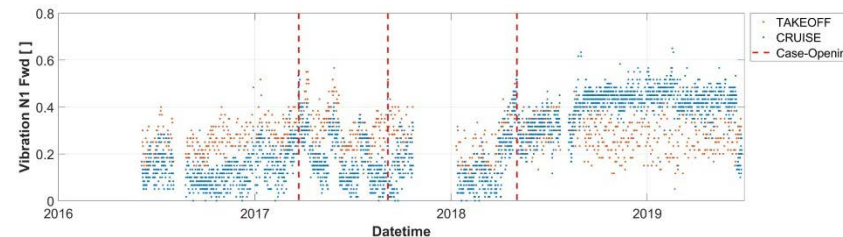
EXAMPLES FROM THE REAL WORLD

Real world data!

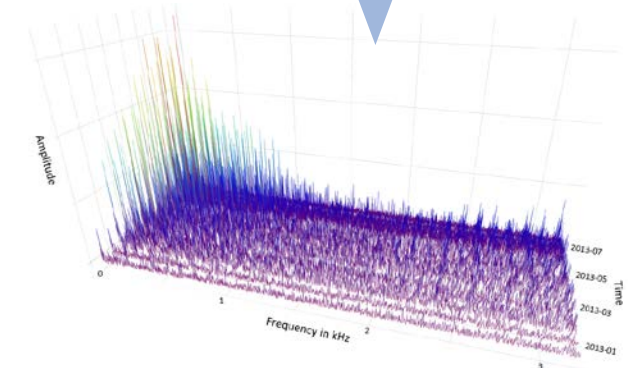
Challenges? Not always the best feature



- Only temperature (-ratio) as feature



- Vibration/temperature/speed signals, but filtered down to singular data points



- Vibration sampled with 40kHz at multiple positions, constant intervals, almost constant measurement conditions

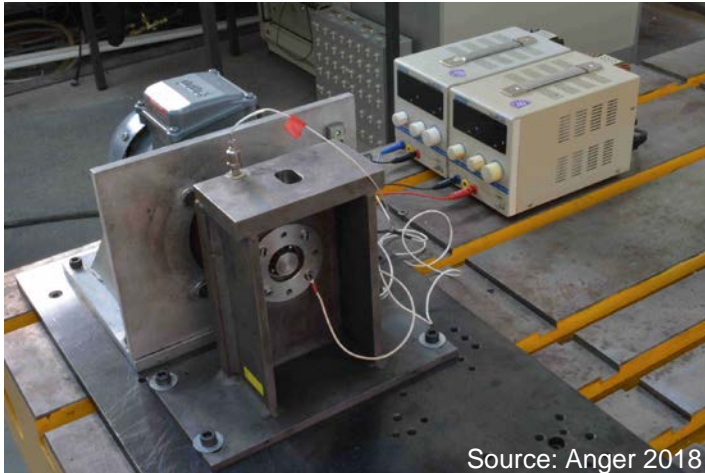
Challenges when handling real world data

- High class imbalance (more than 95 % „healthy“ samples)
- Unexpected/unkown changes of components
- Varying environment conditions
- Varying operational conditions and settings
- Changes of components before failure (no run-to-failure)

Vibration measurement of a degrading gearbox

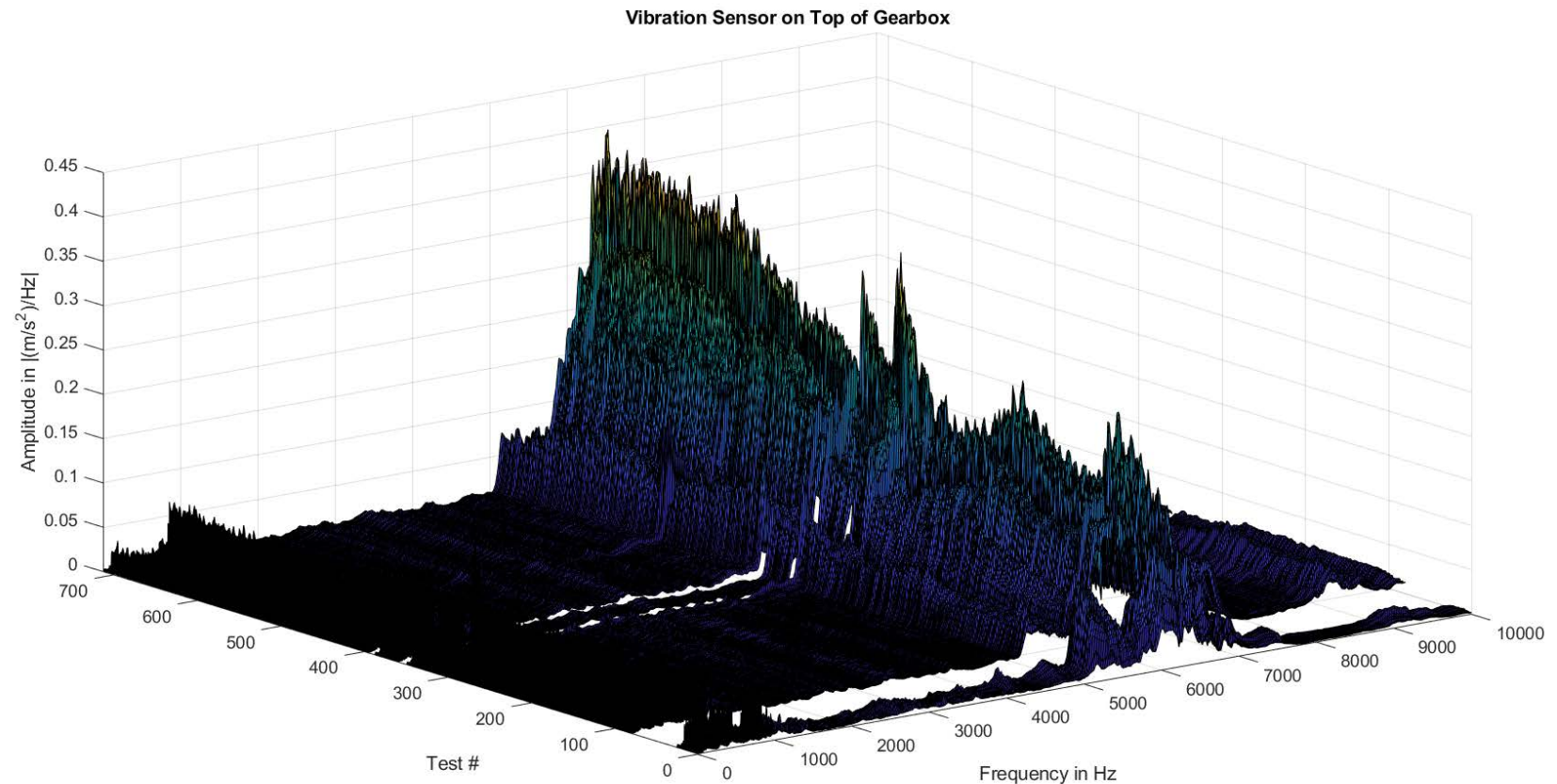
PREVIEW FOR NEXT WEEK

Gearbox health assessment (1/3)



Fast Fourier Transformation

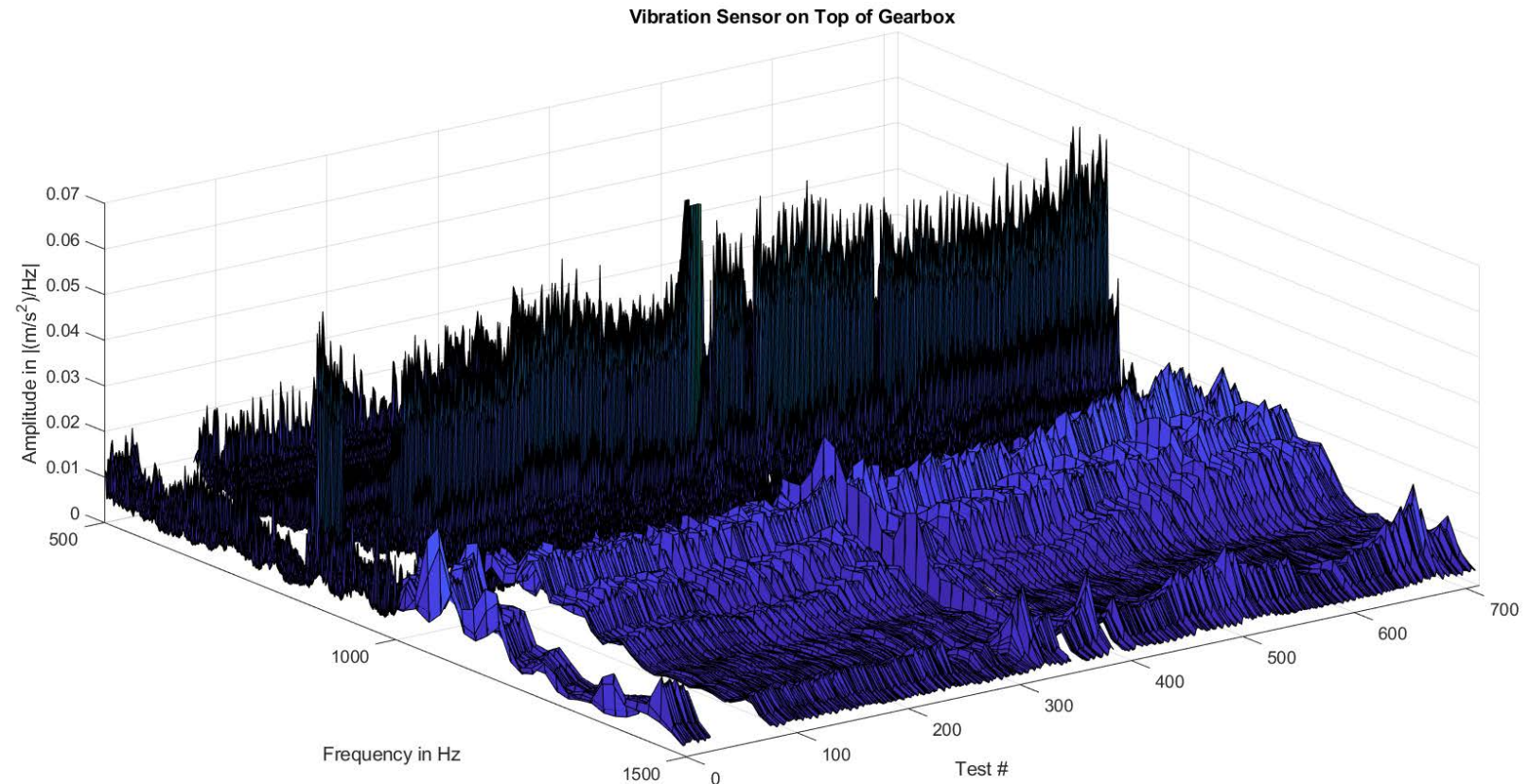
- Whole spectral range can be misleading
 - Process parameter overlie the spectrum
 - High frequencies might originate from subcomponents



Gearbox health assessment (1/3)

Fast Fourier Transformation

- Ranges with high amplitudes do not necessarily include meaningful features
 - Process parameter overlies the spectrum
 - High frequencies might originate from subcomponents

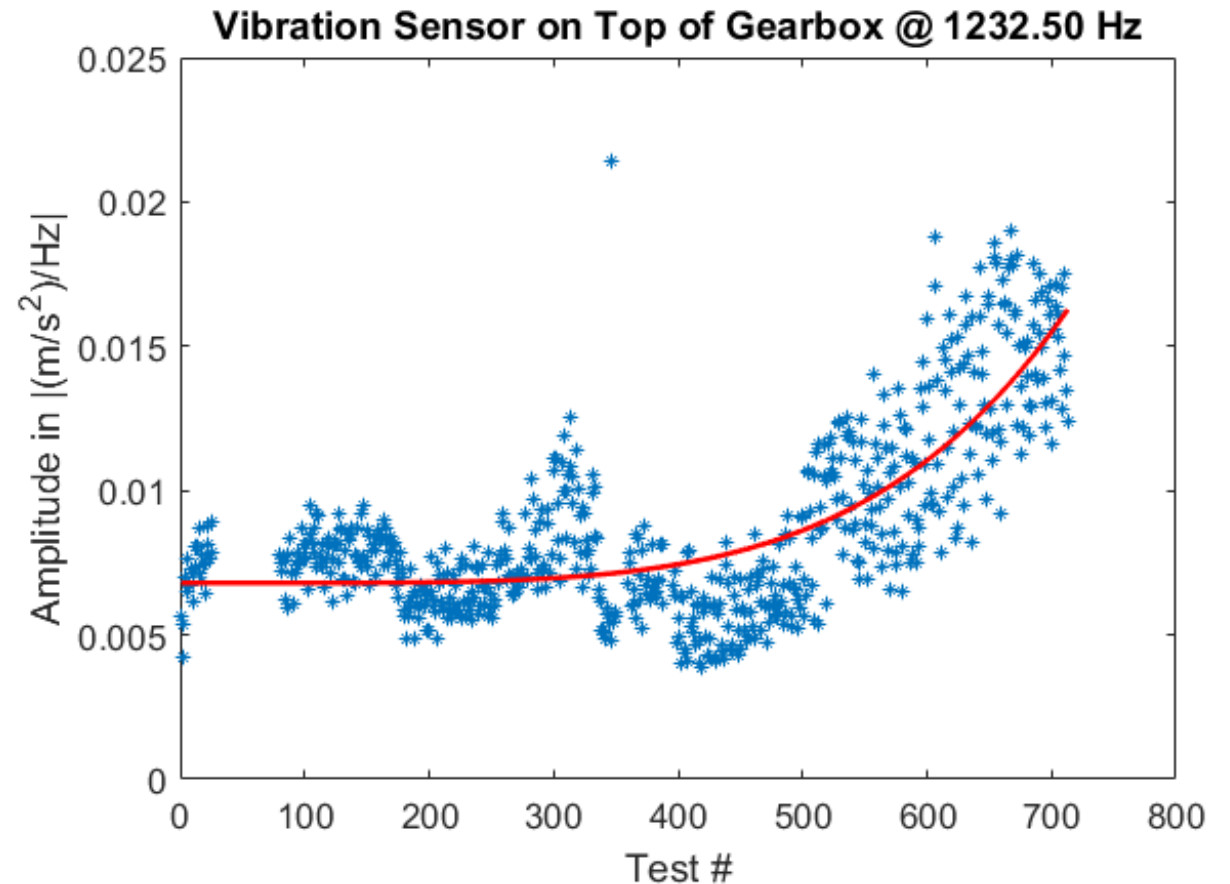


Gearbox health assessment (1/3)

Distinct frequency ranges instead of whole range

- Can be used as features for health assessment or prognosis of RUL
- Fit of the exponential function:

$$HI(t) = a^{\lambda t}$$



What to take with you?

LEARNING OUTCOMES

Key Findings

- **CRISP-DM:** Evaluation has to deal with algorithmic performance and business case fulfillment
- **Data mining / data understanding:**
 - intuitive and easy way to communicate data association rules with apriori algorithms when you need more insight
 - Which predictors are best suited for subsequent analyses must always be checked (manually!) on a case-by-case basis! → grey box modeling
- **Development of model understanding:**
 - comparison of different machine learning models is obligatory
 - Design principle negative learning
- **Model assessment:**
 - Getting a sense for insufficient data quality, overfitting and underfitting by assessing complex problems in several ways
 - One metric is often not enough (except for training) → discussion of multiple model performance metrics

References

- Kim, Nam-Ho, Dawn An, and Joo-Ho Choi. *Prognostics and health management of engineering systems: An introduction*. Springer, (2016)
- Saxena, A., Goebel, K., Simon, D., & Eklund, N.. *Damage propagation modeling for aircraft engine run-to-failure simulation*. In Prognostics and health management (2008)
- Anger, C.. *Hidden semi-Markov Models for Predictive Maintenance of Rotating Elements*. Phd Thesis, TU Darmstadt (2018)
- J. Schaab, *Trusted Health Assessment of Dynamic Systems Based on Hybrid Joint Estimation* (2010)
- I. D. Dinov, *Data Science and Predictive Analytics*. Cham, Springer, 2018.
- L. Breiman, J. Friedman, C. J. Stone and R. A. Olshen, "Random Forest", *Machine Learning*, Vol. 45, Iss. 1, pp. 5-32, 2001
- Breiman, L.; Friedman, J.; Stone, C. J.; Olshen, R. A.: *Classification and Regression Trees*. Taylor & Francis Ltd., London (1984)
- Moisen, G. G.: *Classification and Regression Trees*. Encyclopedia of Ecology, Vol. 1, 582-588 (2008)
- A. Nuic, "User Manual for the Base of Aircraft Data (BADA) Revision 3.12", EUROCONTROL Experimental Centre, Bretigny-sur-Orge, France, EEC Technical/Scientific Report No. 14/04/24-44, August, 2014 [Online] Available: <https://www.eurocontrol.int> [Accessed July 20, 2017]
- Runkler, T. A.: *Data Mining*. Springer Vieweg, Wiesbaden (2015)
- Witten, I. H.; Frank, E.; Hall, M. A.: *Data Mining*. Morgan Kaufmann [Hrsg.]. Elsevier Inc., Burlington (2011)
- Field, A.: *Discovering Statistics Using IBM SPSS Statistics*. SAGE Publications Inc., London (2015)
- Alpaydin, E.: *Maschinelles Lernen*. Oldenbourg Wissenschaftsverlag, München (2008)
- Goodfellow, I.; Bengio, Y.; Courville, A.: *Deep Learning*. MIT press (2016)
- Nunes da Silva, I.; Hernane Spatti, D.; Andrade Flauzino, R.; Bartocci Liboni, L. H.; Franco dos Reis Alves, S.: *Artificial Neural Networks*. Springer (2017)
- Ross, S. M.: *Statistik fuer Ingenieure*. Elsevier, Muenchen (2006)
- McCaffrey, J.: *How To Standardize Data for Neural Networks*. Visual Studio Magazine, 01/2014 (2014)
- Quinlan, J. R.: *Learning With Continuous Classes*. In: Adams & Sterling (eds.) *Proceedings AI 1992*, pp. 343-348. World Scientific, Singapore (1992)
- Flach, P.: *Machine learning: the art and science of algorithms that make sense of data*. Cambridge University Press, Cambridge (2012)
- Sturm, M.: *Neuronale Netze zur Modellbildung in der Regelungstechnik*. Technische Universitaet Muenchen, Institut f`ur Informatik. <https://mediatum.ub.tum.de/doc/601680/601680.pdf> (2000). Accessed 5 January 2018

Contact



DATAbility GmbH
Karlsbader Str. 10, 64295 Darmstadt
Office: Dolivostr. 11, 64293 Darmstadt
www.datability.ai

Sebastian Baumann
baumann@datability.ai

