EMAS
Engineering Modelling, Analysis & Simulation
THE NAFEMS JOURNAL

NAFEMS
Vol. 2, Issue 1, 2025

Robustness investigation of cross-validation based quality measures for model assessment

Thomas Most[a], Lars Gräning[a], Sebastian Wolff[b]

[a]Ansys Germany GmbH, Steubenstraße 25, 99423 Weimar, Germany
[b]Ansys Austria GmbH, Wienerbergstraße 11/A10, A–1100 Vienna, Austria

Abstract

In this paper the accuracy and robustness of quality measures for the assessment of machine learning models are investigated. The prediction quality of a machine learning model is evaluated model-independent based on a cross-validation approach, where the approximation error is estimated for unknown data. The presented measures quantify the amount of explained variation in the model prediction. The reliability of these measures is assessed by means of several numerical examples, where an additional data set for the verification of the estimated prediction error is available. Furthermore, the confidence bounds of the presented quality measures are estimated and local quality measures are derived from the prediction residuals obtained by the cross-validation approach.

Keywords

Machine learning, model prediction, error estimator

1 Introduction

Nowadays, the application of mathematical surrogate models plays an important role in engineering design. Starting with classical Design of Experiment schemes and classical polynomial response surface models [1], [2], meanwhile a wide range of surrogate models has been developed such as Kriging [3], Moving Least Squares [4], Radial Basis Functions [5] and Support Vector Machines [6]. Recently, artificial neural networks [7] have been extended to more sophisticated Deep Learning models [8] which can be applied to a very wide range of engineering fields [9]. A good overview of current applications of surrogate models in global optimization is given in [10] and recent developments in surrogate-assisted global sensitivity analysis can be found in [11]. Investigations on the accuracy of machine learning models for uncertainty quantification are published in [12], [13]. Further reviews on engineering applications are available in [14], [15], [16].

Generally, the application of surrogate models will introduce an additional model error in the prediction. Dependent on the application, the assessment of the approximation quality and the verification of the surrogate model with unknown data is very important as discussed in [17], [18]. A quite common approach for this purpose is the well-known cross-validation [19]. Further methods on model assessment are discussed in [20], [21], [22] where mainly re-sampling methods are considered. A different approach is Bayesian model assessment [23], [24], [25] where the model evidence due to the model parameter uncertainty is evaluated.

In our study we consider quality measures based on cross-validation due to the straight-forward implementation and clear interpretation of the results as discussed recently in [11], [26], [27], [28]. Based on the cross-validation procedure the approximation errors of unknown data points can be estimated. In [29] a variance-based quality measure, the Coefficient of Prognosis (CoP) was introduced

[1]Corresponding author.
E-mail address: thomas.most@ansys.com (T. Most)
https://doi.org/10.59972/f5yl4dl2

based on this principle. With help of this measure a model independent assessment and selection is possible which was realized in the Metamodel of Optimal Prognosis (MOP) in [29] and extended for deep-learning models in [30].

In this paper, the robustness and stability of these quality measures by using different cross validation procedures are investigated. Based on the prediction residuals, the confidence bounds of the CoP are estimated and verified by means of several numerical examples. Additional to the global quality measures, a local model independent error estimator is introduced, which can be utilized for local model improvement by additional samples. Finally, we recommend an extension of the CoP for non-scalar outputs, which is investigated by a further example.

2 Quality measures for the model assessment

2.1 Measuring the goodness of fit

Let us assume a simulation model with a certain number of scalar outputs. Each of these outputs can be represented as a black-box function of a given number of inputs

$$y(x) = f(x_1, x_2, \dots, x_m). \tag{1}$$

If these output functions are approximated by a mathematical surrogate model, we obtain an approximation of the true function

$$\hat{y}(x) = \hat{f}(x_1, x_2, \dots, x_m). \tag{2}$$

If the approximation model is built or trained based on a given number of support points n, we can calculate the residuals for each of the support points and estimate different error measures to quantify the goodness of fit

$$\epsilon_i = y(x_i) - \hat{y}(x_i) = y_i - \hat{y}_i. \tag{3}$$

One well known measure is the root mean squared error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{4}$$

which has the same unit as the output itself and can be interpreted as the standard deviation of the approximation error. Another well-known measure is the unitless Coefficient of Determination (CoD), which measures the ratio of the explained vs. the original variation of the investigated response. In [31] different formulations for CoD are discussed. The two most common formulations are

$$CoD_1 = 1 - \frac{SS_E}{SS_T}, \quad CoD_2 = \frac{SS_R}{SS_T}, \tag{5}$$

where the sum of squared errors SS_E quantifies the unexplained variation, the explained sum of squares SS_R quantifies the explained variation and the sum of total squares SS_T is equivalent to the total variation of the response

$$SS_E = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \quad SS_R = \sum_{i=1}^{n}(\hat{y}_i - \mu_Y)^2, \quad SS_T = \sum_{i=1}^{n}(y_i - \mu_Y)^2, \quad \mu_Y = \frac{1}{n}\sum_{i=1}^{n} y_i. \tag{6}$$

Only for a linear least-squares model, the two formulations in Equation 5 agree and the following equation is valid

$$SS_T = SS_E + SS_R, \quad 0 \le CoD_{1,2} \le 1. \tag{7}$$

The application of the CoD for non-linear models is possible but requires special attention as discussed in [31]. Figure 1 shows an illustrative example, where a quadratic function is approximated with a linear regression model with increasing polynomial order. In this example the synthetic data points contain a small amount of random noise. The figure indicates that a high-order polynomial will tend to fit through the noisy data points and the corresponding SS_E decreases. With increasing approximation order, the non-linearity of the polynomial model and thus the difference between the two formulations of the CoD will increase. The explained sum of squares SS_R could exceed the total sum of squares SS_T and the second formulation of the CoD could lead to values larger than one.

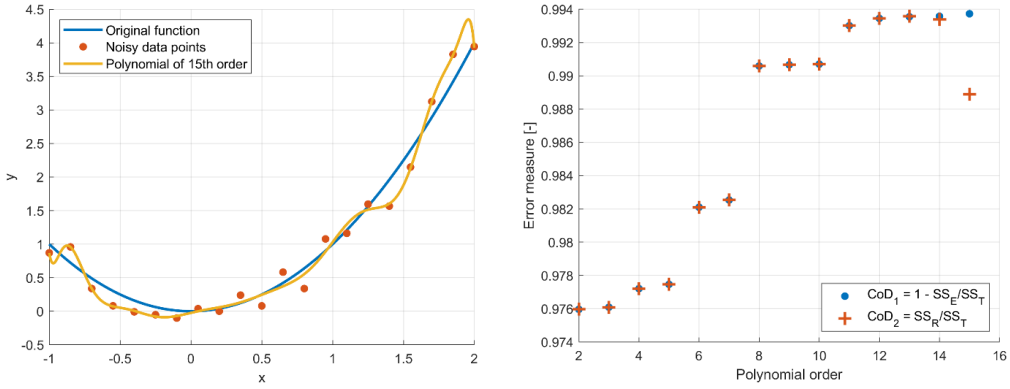

Figure 1. Approximation of noisy data points of a one-dimensional quadratic function with a polynomial model with increasing order.

The first formulation of the CoD could be directly formulated in terms of the squared RMSE as follows

$$CoD = 1 - \frac{SS_E}{SS_T} = 1 - \frac{n^2}{SST} \cdot RMSE^2. \tag{8}$$

This formulation is always smaller or equal one and could be interpreted as the scaled error variance of the approximation model. Only if the sum of squared errors SS_E is larger as the total sum of squares SS_T, the formulation in Equation 8 could be negative. This is not the case for the most approximation models mentioned in the introduction as long a constant baseline is included in the model, which is the case in linear regression [2], Moving Least Squares [4] and Ordinary Kriging [18].

Since the formulation in Equation 8 is directly related to the RMSE, we use this measure in the following paper equivalently to the RSME in order to quantify the deviation between the support point values used for the training and the approximation values at these points. Unfortunately, this will not give us any information of the prediction quality of the surrogate model for unknown data points. Therefore, we extend this measure in the following section.

2.2 Measuring the prognosis quality

In order to estimate the prediction error of a mathematical surrogate model, we can split the data set in two data sets of same size and use set number one for the training and set number two for the estimation of the prediction errors. In a second step this procedure is applied by using data set two for the training and data set one for the estimation. This procedure as shown in Figure 2 is called cross-validation and is explained in more detail in [18].

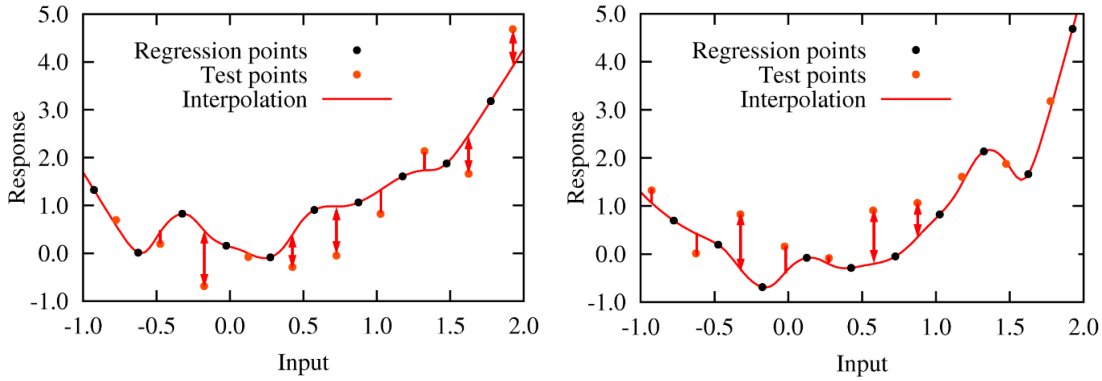

Figure 2. Basic cross-validation procedure by splitting the data set in two subsets: Using set one for training and set two for prediction (left) and set two for training and set one for prediction (right).

More generally, we can subdivide the original data set in q subsets of almost equal size, where the points in each subset should be selected in that way that they cover the investigated space of the input variables almost uniformly. Thus, each of the n support points are mapped to one subset

$$\zeta: \{1, \dots, n\} \rightarrow \{1, \dots, q\}. \tag{9}$$

Once, the q individual cross-validation models have been trained, we use the approximation values to evaluate the prediction residuals for each of the available data points

$$\hat{y}^{cv}(x_i) = \hat{f}_{\sim\zeta(i)}(x_i), \tag{10}$$

where $\hat{f}_{\sim\zeta(i)}(.)$ is the approximation model built by using all cross-validation subsets except the one set belonging to the support point i. Usually, 5-10 subsets are used within the cross-validation procedure to obtain stable estimators [18]. This procedure is called k-fold cross-validation. From this prediction the corresponding residuals of the cross-validation prediction errors can be estimated as

$$\epsilon_i^{cv} = y(x_i) - \hat{y}^{cv}(x_i) = y_i - \hat{y}_i^{cv}. \tag{11}$$

Based on the prediction residuals we can estimate the root mean squared error

$$RMSE^{cv} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i^{cv}\right)^2} \tag{12}$$

and the Coefficient of Prognosis [29]

$$CoP = 1 - \frac{SS_E^{cv}}{SS_T}, \quad SS_E^{cv} = \sum_{i=1}^{n}(y_i - \hat{y}_i^{cv})^2. \tag{13}$$

The CoP quantifies the explained variation in the support data points similarly to the CoD, but the prediction errors estimated with the cross-validation procedure are considered instead of the pure fitting residuals.

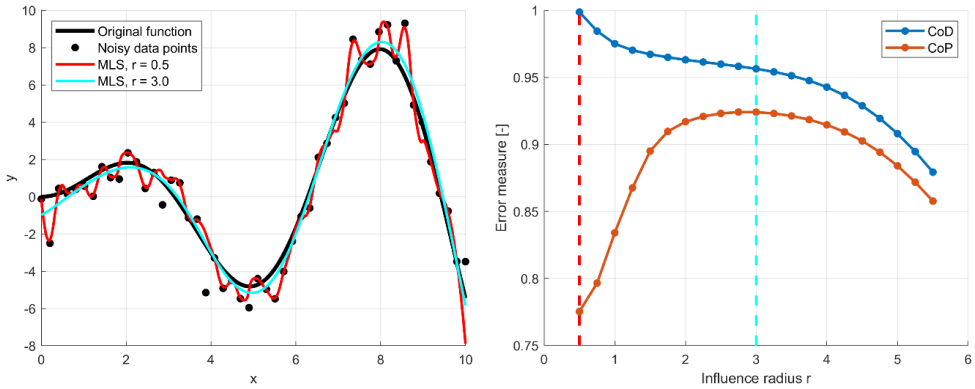

Figure 3. Moving Least Squares approximation of noisy data points of a non-linear function for different values of the influence radius and corresponding difference between CoD and CoP.

In Figure 3 an illustrative example of a one-dimensional non-linear function is given to demonstrate the differences between these two measures. The CoD and CoP are evaluated using Equation 8 and 13 with 5 subsets for the cross-validation procedure. As approximation model a Moving Least Squares (MLS) approximation [4] with quadratic basis is applied. Further details about the implementation can be found in [29]. In Figure 3 the CoD and CoP are shown depending on the influence radius r of the MLS approximation. If the radius is chosen very small, the approximation model will tend to fit well true the noisy support points. As a result, the CoD is close to one. This case is called over-fitting. With increasing influence radius, the approximation model becomes smoother and filters the noise in the supports more efficiently. If the radius is chosen to large, the model will tend to the quadratic basis function and the approximation becomes inadequate. The CoP will indicate a poor model quality if the radius is chosen to small in contrast to the CoD. For a large radius both measures approach to each other and both indicate a poor approximation. The radius with the maximum CoP is the optimal choice for this example and will result in a suitable approximation model as indicated in Figure 3. This simple example shows, that a model evaluation, comparison and possible selection based on the CoP would be much more suitable in order to get the best prediction quality for a given support data set. For models with high flexibility, it could support the appropriate choice of the model parameters in order to prevent over-fitting for noisy data.

The numerical implementation of the k-fold cross-validation procedure is straight-forward for the most classical surrogate models. Our implementation, which is available in the Ansys optiSLang software package [32], considers linear regression, Moving Least Squares, Radial Basis Functions and Kriging with up to 10.000 data points and requires just a small amount of additional numerical effort compared to the model hyper-parameter search. More challenging is the implementation for complex deep learning models, since a re-training for each data subset will not always converge to the same global functions and might stuck in different local optima. To overcome this issue, we developed a specific,

regularized training procedure based on a hybrid approach for the tuning of the optimal network architecture and the evaluation of the prognosis measure. Further details on this approach could be found in [30], [33].

Some mathematical surrogate models provide also closed form solutions for leave-one-out (LOO) cross-validation, where each data set belongs just to a single sample. This so-called leave-one-out (LOO) cross-validation is very attractive from the computational point of view. However, in our examples we will show, that the LOO cross-validation may be too optimistic as an error estimator and the k-fold cross-validation gives more reliable results.

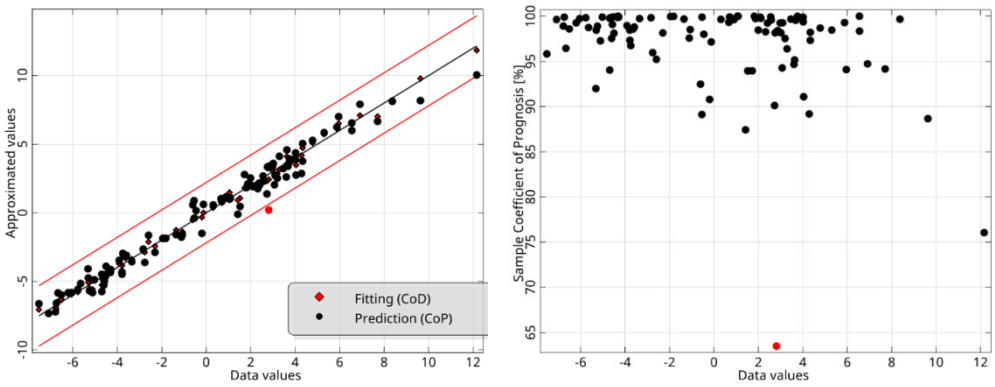

Figure 4. Residual plot with the fitting and prediction residuals (left) and sample CoP (right), which quantifies the contribution of each sample to the CoP.

The residuals of the goodness of fit in Equation 3 and of the cross-validation residuals can be displayed in a so-called residual plot as shown in Figure 4. If a large deviation of the residuals from the fit and the prediction can be observed, we can assume that the applied surrogate model tends to over-fitting. The estimated RMSE in Equation 12 can be used to identify possible outliers. Since the RMSE can be understood as the standard deviation of the approximation error, we can assume a boundary of about $\pm 3 \times RMSE^{cv}$ to check for outliers. In the residual plot in Figure 4, this boundary is indicated as the two red lines. In order to get an estimate on how the residuals of an individual support point x_i contribute to the CoP, we can further formulate the sample CoP as follows

$$CoP_{x_i} = 1 - \frac{(y_i - \hat{y}_i^{cv})^2}{SS_T}, \tag{14}$$

which is shown additionally in Figure 4. The figure clearly indicates that the sample CoP may help to detect outliers more clearly by using a different scaling. The mean value of all individual sample CoPs is consequently the global CoP value introduced in Equation 13.

2.3 Local measures of the prognosis quality

Based on the individual residuals of each support point we can formulate a continuous function of the local prediction error for an arbitrary point in the input space. By using a local averaging scheme similar to the Moving Least Squares approximation [4] the locally weighted RMSE and the local CoP can be formulated as follows

$$RMSE^{cv}(x) = \sqrt{\frac{\sum_{i=1}^{n} w_i(x)\left(y_i - \hat{y}_i^{cv}\right)^2}{\sum_{i=1}^{n} w_i(x)}}, \tag{15}$$

$$CoP(x) = 1 - \frac{\sum_{i=1}^{n} w_i(x)(y_i - \hat{y}_i^{cv})^2}{\sum_{i=1}^{n} w_i(x)} \cdot \frac{n}{SS_T} = 1 - \frac{n \cdot \left(RMSE^{cv}(x)\right)^2}{SS_T}, \tag{16}$$

where $w_i(x)$ is chosen as an exponential, isotropic weighting function, which is scaled with respect to the number of necessary averaging points. In Figure 5 the estimated local prediction errors are shown for the residuals from Figure 4. The figure indicates that in the region of the identified outlier the approximation quality is worst.

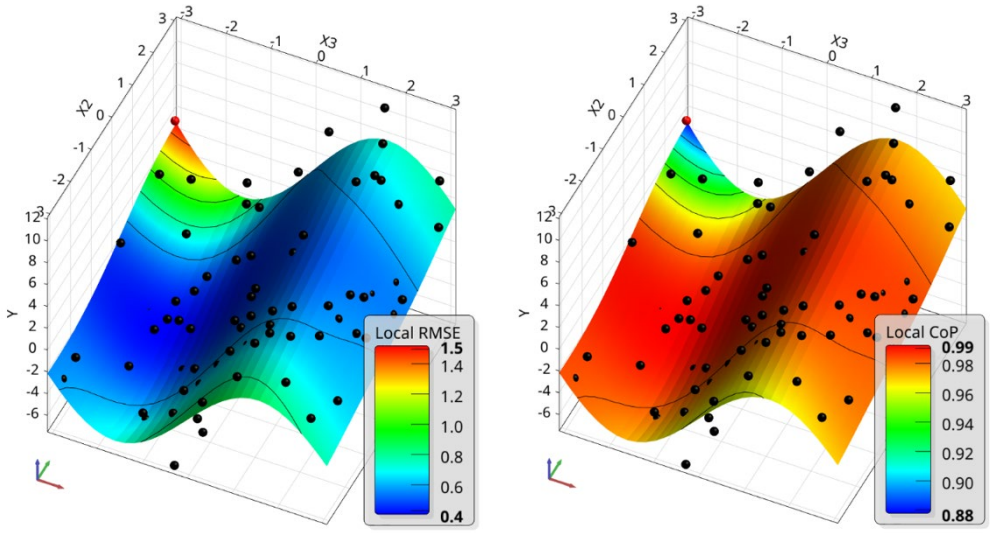

Figure 5. Estimated local root mean squared error (left) and the local Coefficient of Prognosis (right) in a subspace plot of a 5D input space.

The presented local prediction errors can be easily utilized in an adaption scheme such as the expected improvement criterion according to [34]. The advantage of this error estimator is its independence w.r.t. the approximation model type. Thus, it can be applied for simple polynomial models in the same manner as for more sophisticated deep learning networks. This estimator has been applied in the Adaptive Metamodel of Optimal Prognosis (AMOP) [32] in the Ansys optiSLang software package. With help of the local RMSE the prediction uncertainty of an investigated surrogate model can be interpreted as a normally distributed random process, where the mean corresponds to the model approximation itself and the standard deviation to the estimated local RMSE.

2.4 Estimation of confidence bounds using bootstrapping

Once the cross-validation residuals and the prediction quality estimators have been evaluated, one may need further information on the confidence bounds of these estimators. For this purpose, we apply the bootstrapping method introduced in [35]. In this method the statistical properties of an estimator are obtained by sampling from an approximate distribution which can be the empirical distribution of the observed data or a parametrized form of this distribution. In our study we use the most common approach, the non-parametric bootstrapping, where the sampling is done directly from the empirical distribution of the original observations. This method assumes independent and identically distributed observations and constructs a number of re-samples from the original samples. In [36] this procedure is discussed in detail for the estimation of statistical moments of material properties.

In our study we assume the cross-validation residuals of the approximation function in Equation 11 as independent observations of an unknown random number. From this original set of observations $\epsilon_1^{cv}, \epsilon_2^{cv}, \dots, \epsilon_n^{cv}$ a bootstrap sample set $B_j = \epsilon_{1,j}^*, \epsilon_{2,j}^*, \dots, \epsilon_{n,j}^*$ with n samples is selected by random sampling with replacement from the observation data set as illustrated in Figure 6.

In this set each observation ϵ_i^{cv} may appear once, more than once or not at all. This procedure is repeated with a large number of repetitions and the presented model quality measures are estimated for each bootstrap sample set $\boldsymbol{B_j}$ as follows

$$RMSE_{B_j} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\epsilon_{i,j}^*\right)^2}, \quad CoP_{B_j} = 1 - \frac{\sum_{i=1}^{n}\left(\epsilon_{i,j}^*\right)^2}{SS_T}. \tag{17}$$

From the individual results of each bootstrap set $\boldsymbol{B_j}$, the statistical properties of the RMSE and CoP estimates can be evaluated.

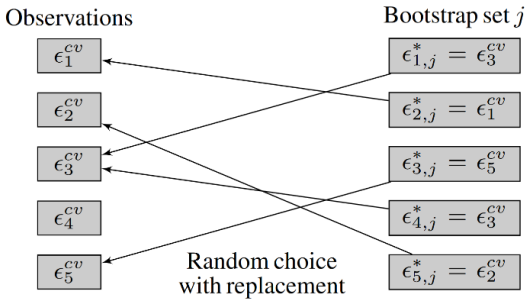

Figure 6. Principle of non-parametric bootstrap method: generation of a bootstrap sample set from the original residual set by random choice with replacement.

In Figure 7 the 100 cross-validation residuals of the previous example plots are shown. The anthill plot indicates an almost independent relation between the data values and the residuals. However, the histogram is non-symmetric and indicates a skewed distribution. For these residuals the bootstrap resampling is applied using 10^5 repetitions and the statistical measures are evaluated for each of the bootstrap samples. In Figure 7 the histograms of the corresponding RMSE and CoP are shown including the 99% confidence intervals, which can be directly estimated from the bootstrap samples. The figure indicates an almost symmetric distribution of the RMSE, which would fit to a normal distribution very well. The distribution of the CoP is non-symmetric and skewed, which means that the mean value and the standard deviation might be not sufficient to characterize the confidence interval. Therefore, we calculate the confidence intervals of each quality directly from the bootstrap samples without assuming any distribution.

The benefit in bootstrapping the residuals directly instead of training new surrogate models for each bootstrap set is clearly the reduction of the numerical effort. Once the cross-validation residuals are obtained for a given support point set, the bootstrapping and the evaluation of the CoP distribution can be performed very cheap. However, the estimator will not cover the case, that the support points do not have a suitable distribution. Nevertheless, the confidence estimates from this procedure are quite helpful to assess the quality estimators as shown in the numerical examples.

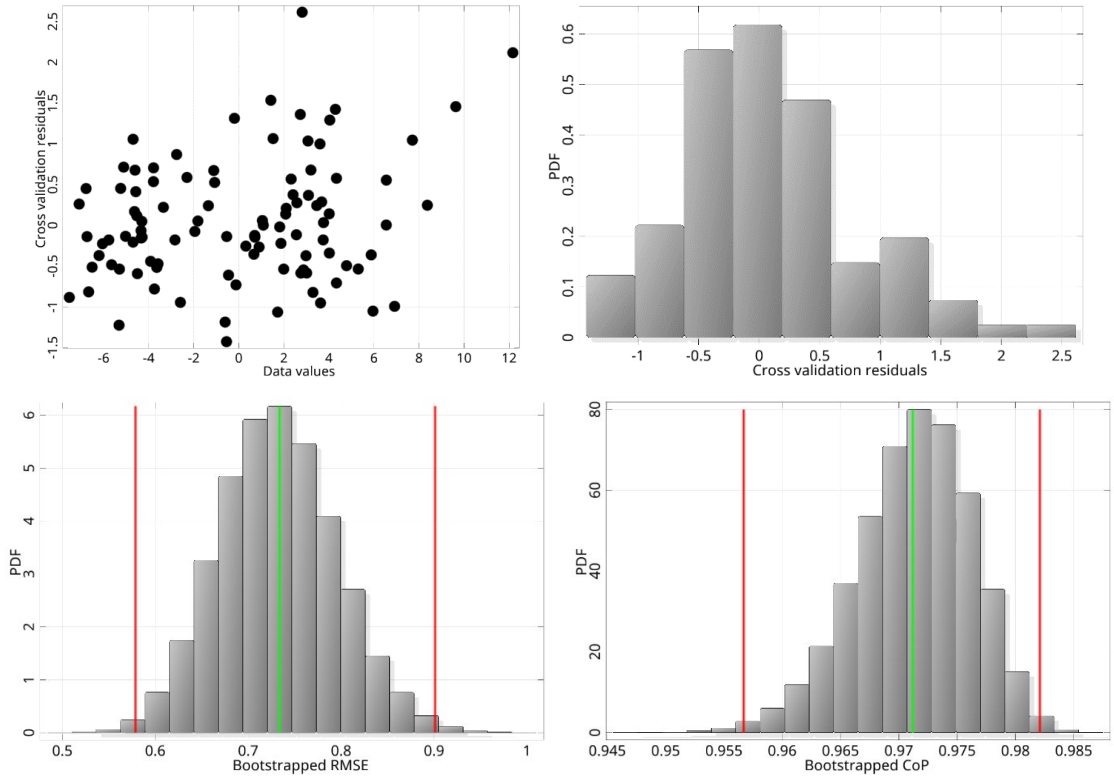

Figure 7. Cross-validation residuals of 100 support points: distribution and histogram (top) and bootstrapped RMSE and CoP (bottom) with deterministic estimates (green) and 99% confidence interval (red).

2.5 Extension to non-scalar outputs

The non-scalar outputs of a simulation model can be described as a function of the vector of input parameters $\mathbf{x}$ and a discretization vector $\mathbf{t}$

$$y(x,t) = f(x,t). \tag{18}$$

This discretization could be defined by a specific time step of a time-series output, a spatial coordinate of the stress or strain field outputs of a finite element model or a combination of spatial and time discretization. Let us assume, that the discretization vector maps a spatial output object to a single scalar output as shown in Figure 8 for a one-dimensional time-series output.

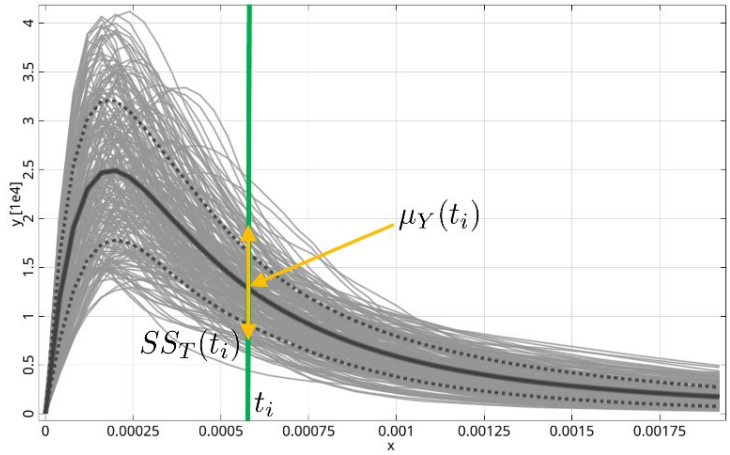

Figure 8. Samples of a time-series and indicated mean and sum of total squares at a certain discretization point t_i.

If the mapping of the discretization is unique for every sample, we can formulate the residuals of the simulation and the approximation model as follows

$$\epsilon_j(t_i) = y(x_j,t_i) - \hat{y}(x_j,t_i) = y_j(t_i) - \hat{y}_j(t_i). \tag{19}$$

With this formulation we can introduce the spatial sum of errors and sum of squares accordingly to the scalar outputs

$$SS_E(t_i) = \sum_{j=1}^{n} \left(y_j(t_i) - \hat{y}_j(t_i)\right)^2, \quad SS_T(t_i) = \sum_{j=1}^{n} \left(y_j(t_i) - \mu_Y(t_i)\right)^2, \quad \mu_Y(t_i) = \frac{1}{n}\sum_{j=1}^{n} y_j(t_i). \tag{20}$$

The discretized formulations for SS_E and SS_T could be used to calculate the CoD accordingly to Equation 8, which would normalize the output residuals at each discretization point individually

$$CoD(t_i) = 1 - \frac{SS_E(t_i)}{SS_T(t_i)}. \tag{21}$$

If a stress field or a time-series has small variations in a certain region this normalization might be difficult to interpret since it might indicate low CoD values for similar SS_E estimates just due to the different normalization with $SS_T(t_i)$. If the spatial or time-series output is assumed to be a stationary random process, the stationary CoD might be a more suitable measure for this type of applications

$$CoD^{stat}(t_i) = 1 - \frac{SS_E(t_i)}{SS_T^{stat}}, \tag{22}$$

where the stationary sum of squares could be assumed as the stationary variance of the whole non-scalar output considering n_d discretization points

$$SS_T^{stat} = \frac{1}{n_d}\sum_{i=1}^{n_d}\sum_{j=1}^{n} \left(y_j(t_i) - \mu_Y(t_i)\right)^2 = \frac{1}{n_d}\sum_{i=1}^{n_d} SS_T(t_i). \tag{23}$$

Similar to the stationary CoD, we can define the stationary CoP as follows

$$CoP^{stat}(t_i) = 1 - \frac{SS_E^{cv}(t_i)}{SS_T^{stat}}, \quad SS_E^{cv}(t_i) = \sum_{j=1}^{n} \left(y_j(t_i) - \hat{y}_j^{cv}(t_i)\right)^2. \tag{24}$$

The cross-validation procedure and the calculation of the residuals is straight-forward from the mathematical viewpoint, similar as for scalar outputs. However, the residuals require a unique mapping to a reference discretization. Nevertheless, the analysis of finite element meshes with high resolution requires an efficient implementation of the cross-validation procedure and especially the sensitivity estimation. In [37], [38], [39], [40] further details on different approximation models and discretization types for non-scalar outputs are discussed.

3 Benchmark results and applications

3.1 Analytical benchmark function

In a first example, we investigate an analytical benchmark function with 5 inputs

$$y(x) = 0.5 \cdot x_1 + x_2 + 0.5 \cdot x_1 x_2 + 5.0 \cdot sin(x_3) + 0.2 \cdot x_4 + 0.1 \cdot x_5, \quad -\pi \leq x_i \leq \pi, \tag{25}$$

which is shown in Figure 9 in different two-dimensional subspaces of the inputs while keeping the remaining inputs constant at the mean values. This benchmark function was introduced in [29] and consists of additive linear and non-linear terms and one coupling term. Furthermore, the inputs x_4 and x_5 have minor importance. We investigate this example by generating 50 support points within the input bounds by using an improved Latin-Hypercube Sampling (LHS) according to [41]. An isotropic Kriging approximation model according to [18] is trained by using these support points and a k-fold and LOO cross-validation is performed to estimate the prediction errors. Unimportant inputs are removed automatically from the approximation model using the Metamodel of Optimal Prognosis approach [29]. 500 additional test samples are generated by an independent LHS and are evaluated with the benchmark function. These samples are used to compare the estimated prediction errors from the cross-validation procedure with the errors in unknown data. For this purpose, the prediction sum of squares SS_E is evaluated for the cross-validation residuals and for the additional test data according to Equation 13.

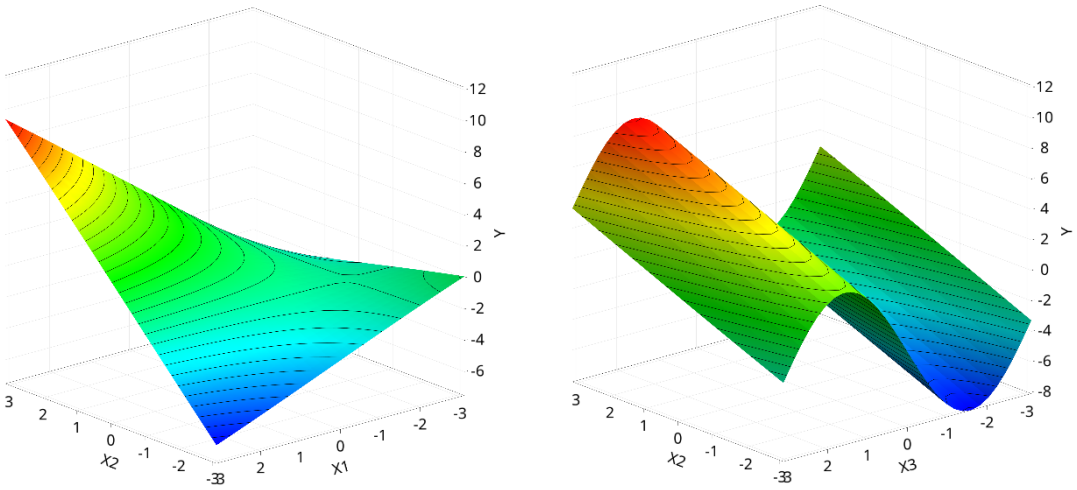

Figure 9. Analytical 5D benchmark function plotted in the 2D subspaces spanned by x_1-x_2 and x_2-x_3.

In order to quantify the statistical scatter of the prediction error estimates, we generate 50 independent data sets for the support points and perform the model training and error estimation and compare these estimates with the prediction error of a fixed test data set. Since the SS_T itself varies for each support point set, we compare not directly the estimated CoP from the cross-validation with the CoD of the test data. Instead, we scale the SS_E^{cv} from the cross-validation with the SS_T of the test data as follows

$$\Delta SS_E^{cv} = \frac{\frac{1}{n} SS_E^{cv} - \frac{1}{n_t} SS_E^{test}}{\frac{1}{n_t} SS_T^{test}}, \tag{26}$$

where the normalization with the number of supports n and the number of test data points n_t is necessary due to the different number of samples in both sets. In Figure 10 the obtained ΔSS_E^{cv} are shown for the 50 investigated runs by using LOO as well as k-fold-cross-validation in the prediction quality estimation. The figure indicates, that in case of LOO the number of runs, where the SS_E is over-

estimated, is similar as the number of cases where the SS_E is under-estimated. If the k-fold-cross-validation is used, the estimated SS_E is mostly larger as verified by the test data, which is indicated by $\Delta SS_E^{cv} > 0$. This means that the CoP estimate is in the most cases more conservative and does not over-estimate the prediction quality of the investigated surrogate model. If the number of support points is increased, the deviation between the LOO and k-fold-cross-validation quality estimates reduces.

Additionally, we investigate the confidence of the estimated CoP compared to the corresponding CoD of the test data. The confidence interval of the CoP is estimated directly from the k-fold cross validation residuals for each run using the bootstrap approach with 10^5 repetitions. In Figure 11 the CoP estimates with 99% confidence bounds are shown for the 50 investigated runs. The figure indicates that the confidence interval of the CoP covers the verified CoD in almost all cases. If we look deeper into the results, we can observe, that for several runs the CoP estimate is similar, but the confidence bounds differ significantly. This is the case for the sorted run numbers 34 and 35. In Figure 12 the residual plots and the histogram of the bootstrapped CoP of both cases are shown. The figure indicates, that for run 34 with the larger confidence interval, one significant outlier can be observed while the remaining residuals are smaller. In run number 35 the residuals indicate no significant outlier but have larger variation as in run number 34. This means, that in case of possible outliers the confidence interval of the CoP should be larger, and a narrow estimate of the CoP is not possible.

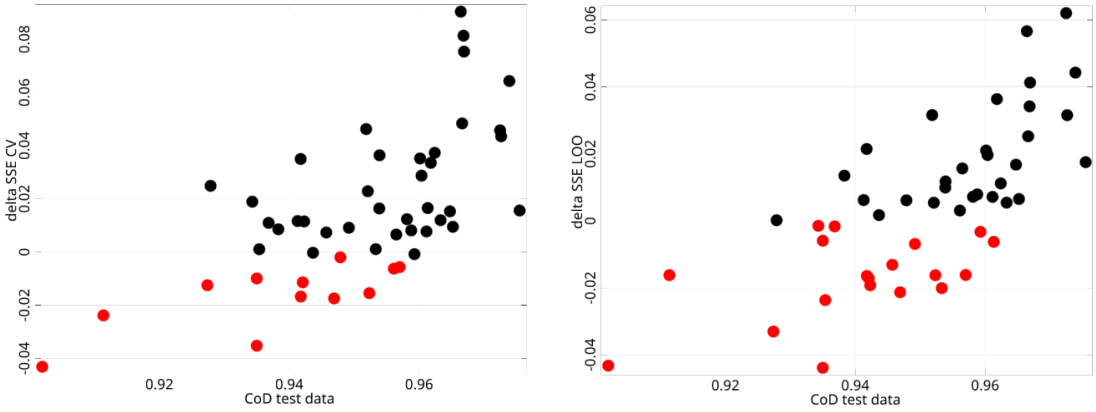

Figure 10. Statistical evaluation of the prediction errors of the analytical test function by using 50 support points and 500 test points with k-fold-cross-validation (left) and LOO-cross-validation (right).

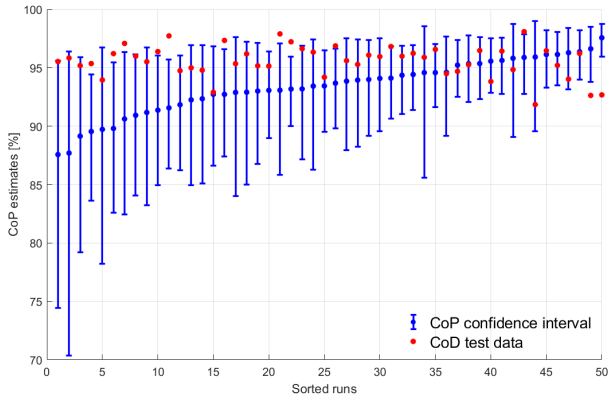

Figure 11. CoP estimates and confidence bounds of the analytical test function by using k-fold cross-validation compared to the CoD of the test data for 50 support points.

3.2 Noisy benchmark function

In the second example we extend the analytical function with additional linear, non-linear and noise terms. The function for 20 inputs reads

$$y(x) = 0.5 \cdot x_1 + x_2 + 0.5 \cdot x_1 x_2 + 5.0 \cdot sin(x_3) + 0.5 \cdot x_4 + 0.5 \cdot x_4^2 + 0.1 \cdot x_5 + \sum_{i=6}^{20} 0.01 \cdot x_i + 0.5 \cdot \mathcal{N}(0,1), \quad -\pi \leq x_i \leq \pi, \tag{27}$$

where $\mathcal{N}(0,1)$ is a standard normal noise term.

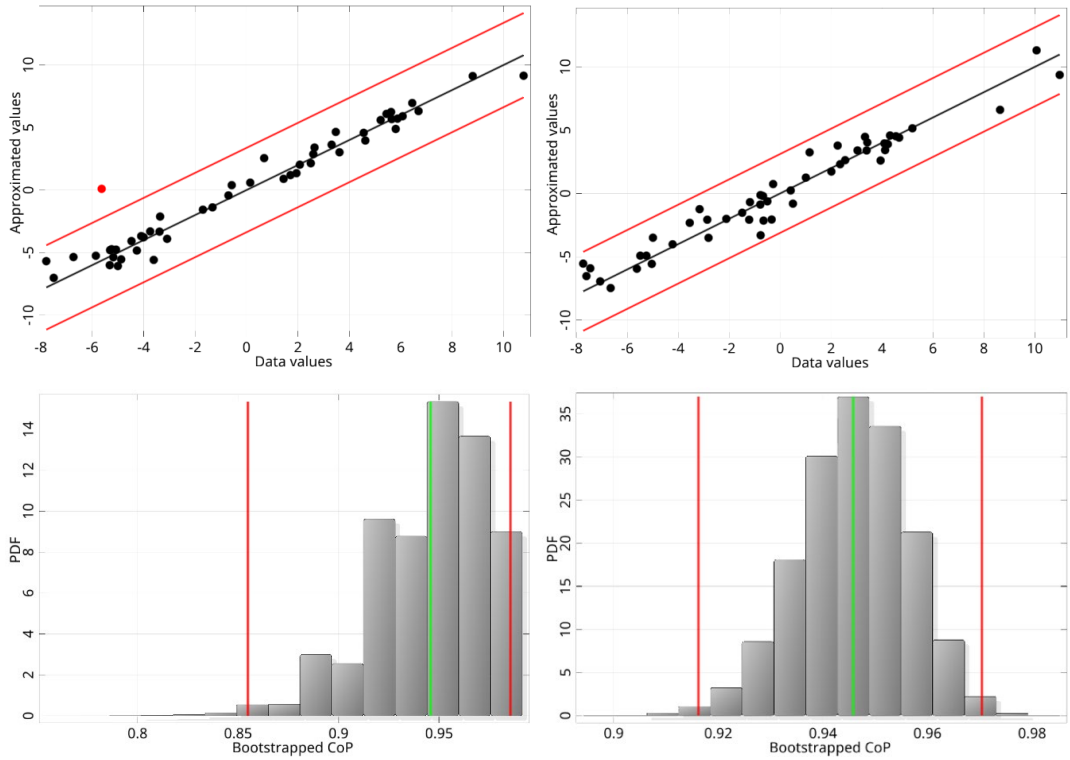

Figure 12. Residual plots and bootstrapped CoP's of the analytical test function of sorted run number 34 (left) and run number 35 (right) by using 50 support points.

We generate 100 support points and 500 additional test samples by Latin-Hypercube Sampling (LHS) and apply an isotropic Kriging approximation model. The scatter of the statistical measures is analyzed again by evaluating 50 runs with k-fold-cross-validation. In Figure 13 the estimated confidence intervals are compared to the CoD of the additional test data. As in the previous example, the estimated confidence bounds of the CoP and the verified CoD agree very well.

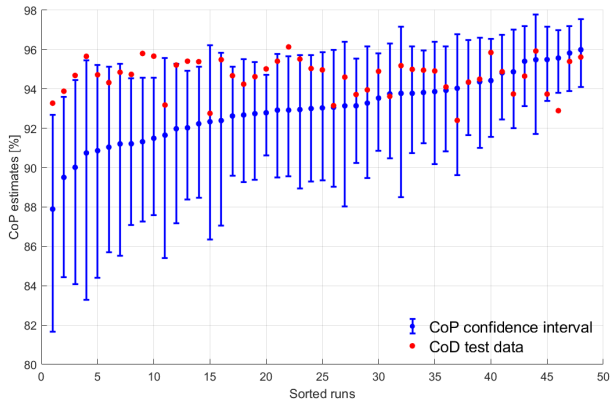

Figure 13. CoP estimates and confidence bounds of the noisy test function by using k-fold-cross-validation compared to the CoD of the test data for 100 support points.

3.3 Front crash example

In the third example we investigate the presented error measures on a highly non-linear application, where the intrusions and pulses of a truck impact example are analyzed with the LS-Dyna finite element solver as shown in Figure 14. The pulses are acceleration related quantities computed over two-time-intervals of the crash event. 22 input variables have been considered in the analysis which belong to the metal sheet thicknesses and the material properties of specific parts of the car body. Further details on this example can be found in [42]. For this example, different data sets of 100, 200 and 400 Latin Hypercube samples have been used for the model training and a single test data set of 1200 samples for the validation of the estimated prediction errors.

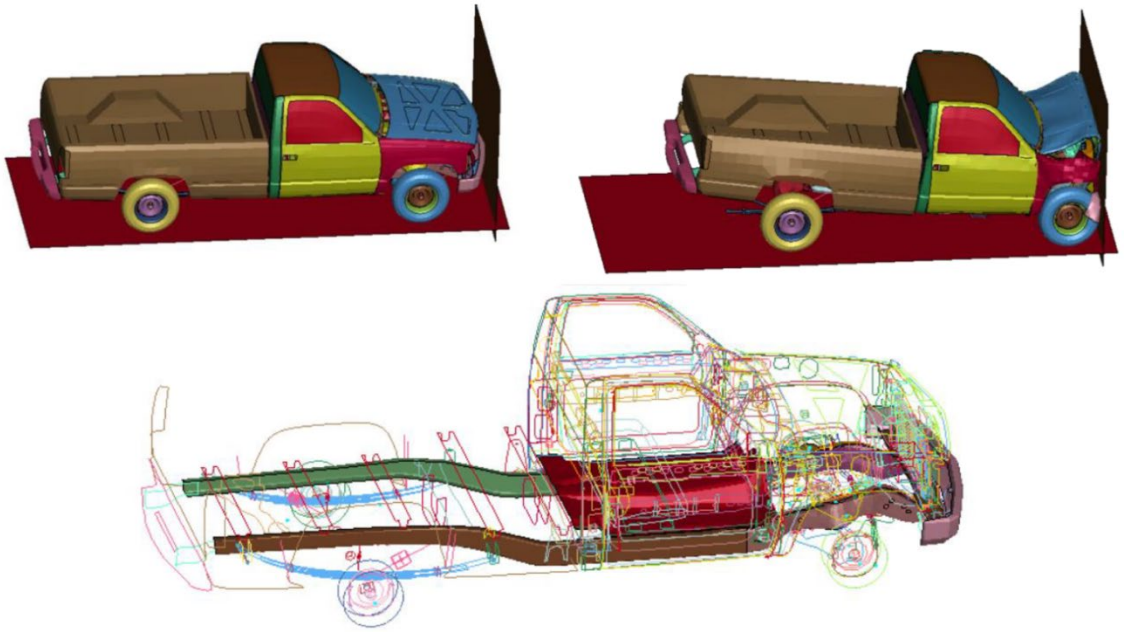

Figure 14. Investigated front crash example according to [42] considering 22 varying inputs of specific parts of the car body in a LS-Dyna simulation model.

Table 1. Computed quality estimates for the front crash example with 22 inputs and 6 investigated outputs by using k-fold cross validation and residual bootstrapping

Output	No. supports	Selected Model	No. selected inputs	CoP	99% conf. interval	CoD test data
N1_disp	100	Linear Polynomial	9	0.747	0.643 - 0.830	0.759
	200	Anisotropic Kriging	14	0.803	0.735 - 0.856	0.809
	400	Anisotropic Kriging	18	0.835	0.793 - 0.867	0.836
N2_disp	100	Linear Polynomial	9	0.778	0.676 - 0.856	0.787
	200	Anisotropic Kriging	14	0.827	0.762 - 0.876	0.836
	400	Anisotropic Kriging	15	0.857	0.823 - 0.885	0.853
Stage1Pulse	100	Anisotropic Kriging	11	0.990	0.986 - 0.993	0.989
	200	Anisotropic Kriging	13	0.992	0.989 - 0.994	0.994
	400	Anisotropic Kriging	13	0.994	0.992 - 0.995	0.994
Stage2Pulse	100	Linear Polynomial	20	0.942	0.922 - 0.961	0.908
	200	Anisotropic Kriging	18	0.956	0.946 - 0.965	0.932
	400	Anisotropic Kriging	19	0.967	0.961 - 0.973	0.954
total_mass	100	Linear Polynomial	9	1.000	1.000 - 1.000	1.000
	200	Linear Polynomial	9	1.000	1.000 - 1.000	1.000
	400	Linear Polynomial	9	1.000	1.000 - 1.000	1.000
Head injury criterion (HIC)	100	Anisotropic Kriging	2	0.062	0.000 - 0.752	0.000
	200	Anisotropic Kriging	19	0.365	0.000 - 0.686	0.000
	400	Anisotropic Kriging	21	0.318	0.000 - 0.677	0.000

Again, we use the Metamodel of Optimal Prognosis [29] to select the most suitable approximation model for each response automatically. As approximation models we consider polynomials and Moving Least Squares, each with linear and quadratic basis, as well as isotropic and anisotropic Kriging. Additional to the best approximation model, the optimal subspace of important inputs is detected by using the maximum Coefficient of Prognosis as selection criterion.

In Table 1 the results for the investigated six responses are given. The table indicates, that with increasing number of support points, the prediction quality estimated with the CoP and verified with the test data set increases for almost all outputs. Furthermore, the estimated confidence interval of the CoP covers the verified test CoD very well. The table further indicates, that with increasing number of supports the number of selected important inputs increases, which is a typical phenomenon in machine learning. For the HIC output, the CoP and its confidence interval indicate a very low prediction quality, which might be caused by numerical noise in the output or a high-dimensional non-linear relation between the inputs and the HIC value.

In Figure 15 the residuals and the bootstrapped CoP's are shown exemplarily for the *N1_disp* displacement response obtained for 400 support points. For this output a clear improvement of the prediction quality can be observed with increasing number of support points, which is indicated by a narrower confidence interval. In the residual plots no significant outlier or systematic approximation errors could be recognized. This is not the case for the HIC value residuals shown in Figure 16. Here a clear systematic approximation error could be detected, which confirms the estimated poor approximation quality. The calculated confidence intervals cover almost the whole domain of possible CoP values.

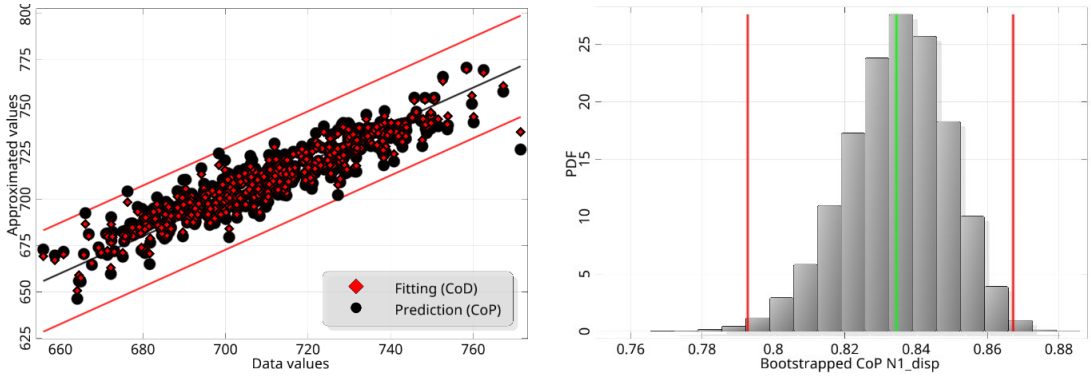

Figure 15. Residual plots (left) and bootstrapped CoP's (right) of the output *N1_disp* from the front crash example by using 400 support points.

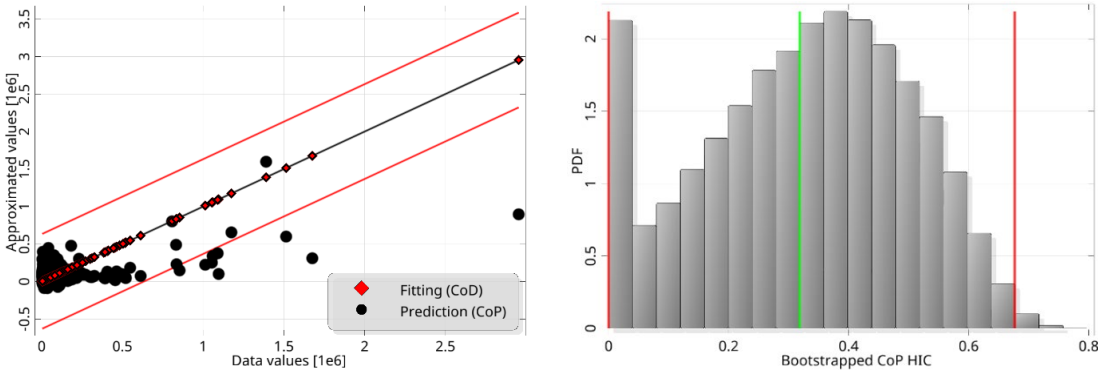

Figure 16. Residual plots (left) and bootstrapped CoP's (right) of the HIC value by using 400 supports.

3.4 Cut-In scenario example

In the following example, the simulation data of a Cut-In scenario of an autonomous vehicle are analyzed. Further details of the simulation analysis can be found in [43]. In this example 10 input parameters as ego and cut-in vehicle speeds, lead vehicle distance and breaking deceleration are considered. In the simulation the typical key performance indicators (KPIs) as critical time headway (THW), time to collision (TTC), collision speed and many others have been calculated. From these outputs a combined failure criterion was derived for each simulation run.

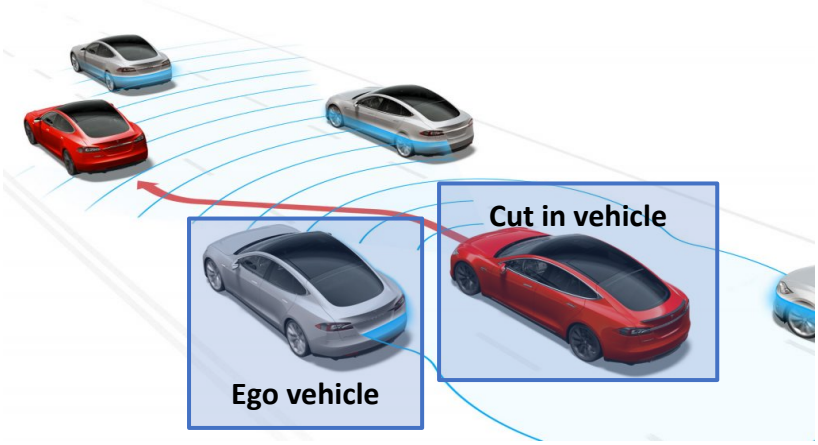

Figure 17. Simulated Cut-Scenario of an autonomous vehicle.

For the analysis of the machine learning models, different data sets of 280, 560, 1120 and 1866 support points have been used for the training and 5600 data points are considered as verification data. Similar as in the previous example, different approximation models have been considered in the MOP competition and the most important inputs have been detected automatically. In Table 2 the estimated CoPs for the training data and the CoDs of the verification data set are given for four selected outputs including the confidence interval from the bootstrapped residuals.

The table shows, that for each investigated output, the estimated CoP increases with increasing number of training points. Furthermore, the CoD of the verification agrees very well with the CoP estimates and the corresponding confidence bounds.

Table 2. Computed quality estimates for the Cut-In example with 10 inputs and 4 investigated outputs by using k-fold cross validation and residual bootstrapping.

Output	No. supports	Selected Model	No. selected inputs	CoP	99% conf. interval	CoD test data
Time Headway (THW)	280	Anisotropic Kriging	7	0.666	0.568 - 0.750	0.745
	560	Anisotropic Kriging	7	0.722	0.659 - 0.776	0.771
	1120	Anisotropic Kriging	7	0.794	0.756 - 0.830	0.804
	1866	Anisotropic Kriging	8	0.824	0.795 - 0.850	0.841
Time to collision (TTC)	280	Anisotropic Kriging	10	0.417	0.112 - 0.661	0.244
	560	Anisotropic Kriging	8	0.459	0.281 - 0.613	0.490
	1120	Anisotropic Kriging	10	0.506	0.385 - 0.613	0.549
	1866	Anisotropic Kriging	9	0.554	0.464 - 0.634	0.600
Ego max speed	280	Anisotropic Kriging	5	0.804	0.735 - 0.863	0.751
	560	Anisotropic Kriging	8	0.792	0.721 - 0.845	0.785
	1120	Anisotropic Kriging	9	0.827	0.791 - 0.858	0.824
	1866	Anisotropic Kriging	8	0.843	0.812 - 0.869	0.837
Criticality	280	Anisotropic Kriging	7	0.704	0.583 - 0.800	0.721
	560	Anisotropic Kriging	8	0.713	0.636 - 0.782	0.758
	1120	Anisotropic Kriging	8	0.773	0.722 - 0.819	0.797
	1866	Anisotropic Kriging	8	0.808	0.772 - 0.840	0.830

In Figure 18 the approximation model for the time-headway output is shown exemplarily for 280 and 1866 training points. In the first case the model already represents the global behavior, but local nonlinearities are filtered. For 1866 training points these local relations can be represented much more accurate. In Figure 19 the corresponding residual plots and the histograms of the bootstrapped CoP of both cases are shown. The figure indicates that even with 1866 training points a perfect approximation of the simulated time headway is not possible. However, the estimated CoP was proven to be an accurate and reliable measure for the model prediction quality.

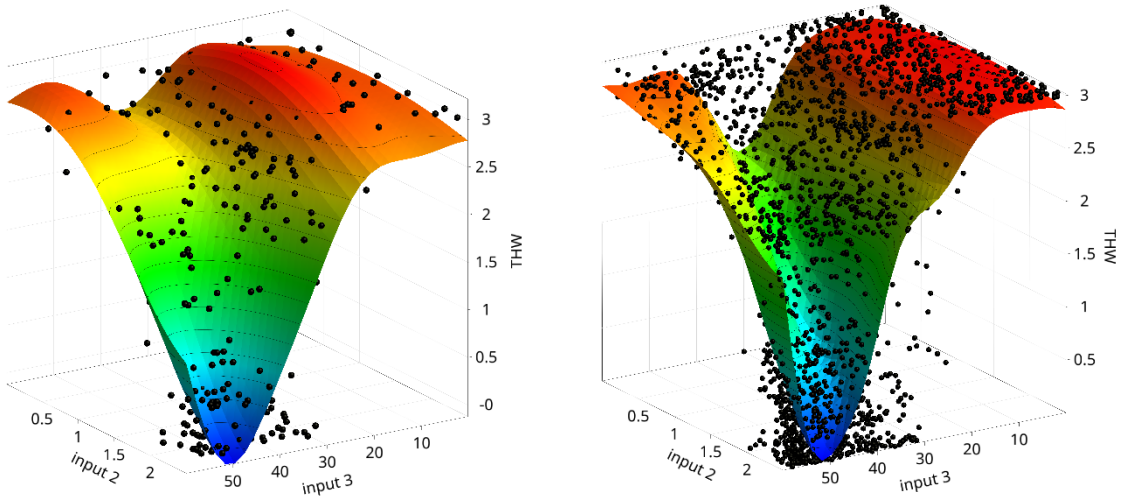

Figure 18. Approximation model of the time headway (THW) for 280 training points (left) and 1866 training points (right) in the subspace of the two most important inputs of the Cut-In scenario example.

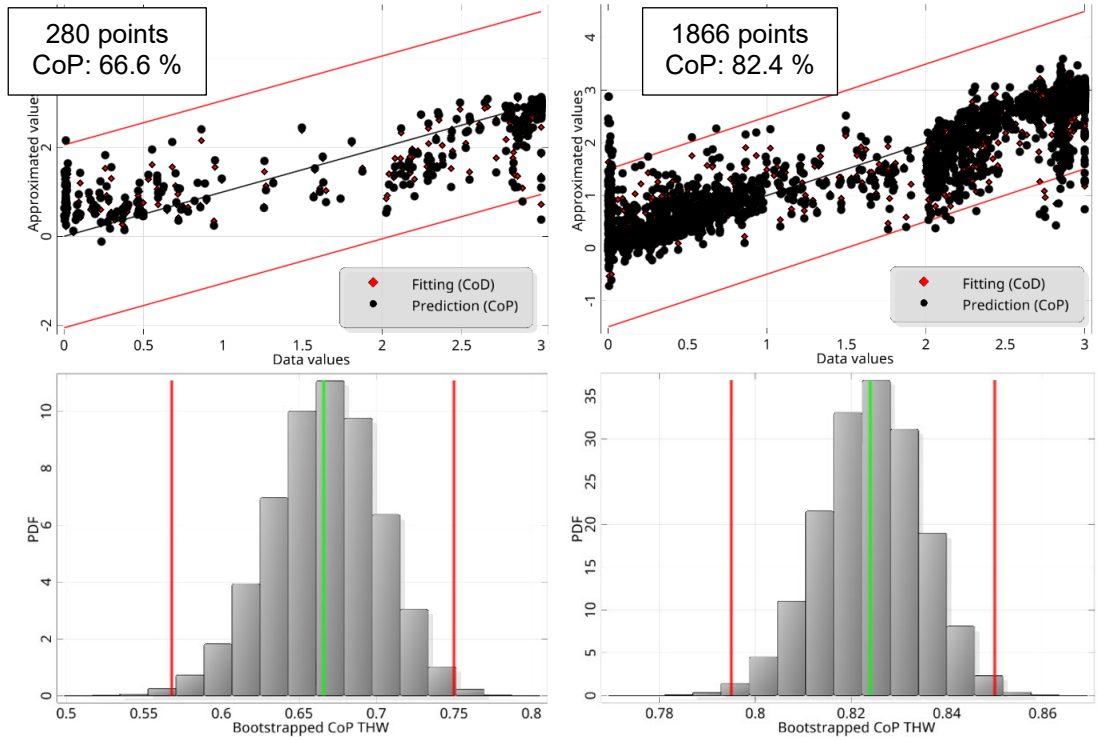

Figure 19. Residual plots (top) and bootstrapped CoP's (bottom) of the time headway output of the Cut-In scenario example.

3.5 Wedge splitting example with non-scalar outputs

In the final example, the CoP estimates are investigated for non-scalar outputs. We consider the load-displacement curve of a wedge splitting test as one-dimensional output. The simulation model shown in Figure 20 considers an elastic base material and a predefined crack with bi-linear softening law. The structure is discretized by 2D finite elements, and the softening curve is obtained by a displacement-controlled simulation. The displacements are measured as the relative displacements between the load application points. Further details on the simulation model can be found in [44]. Six material parameters are varied to generate the samples: the Young's modulus, the Poisson's ratio, the tensile strength, the Mode-I fracture energy and two shape parameters of the bi-linear softening law.

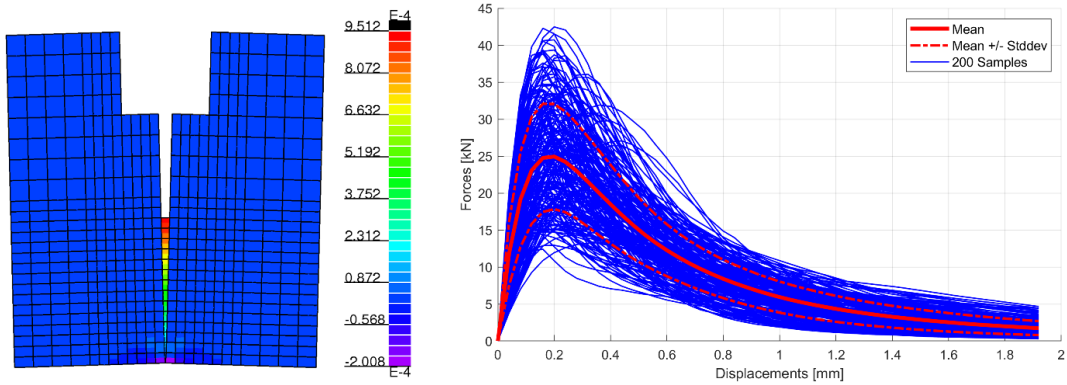

Figure 20. Wedge splitting test example: finite element model (left) and simulated 200 Latin Hybercube samples of the load-displacement curve (right).

The samples of the load-displacement curves are discretized at 49 equidistant displacement points. As approximation model we utilize the Deep Gaussian Covariance Network [39], where a one-dimensional function is represented as a correlated Gaussian process model. A further application of this model for time-series approximation can be found in [40].

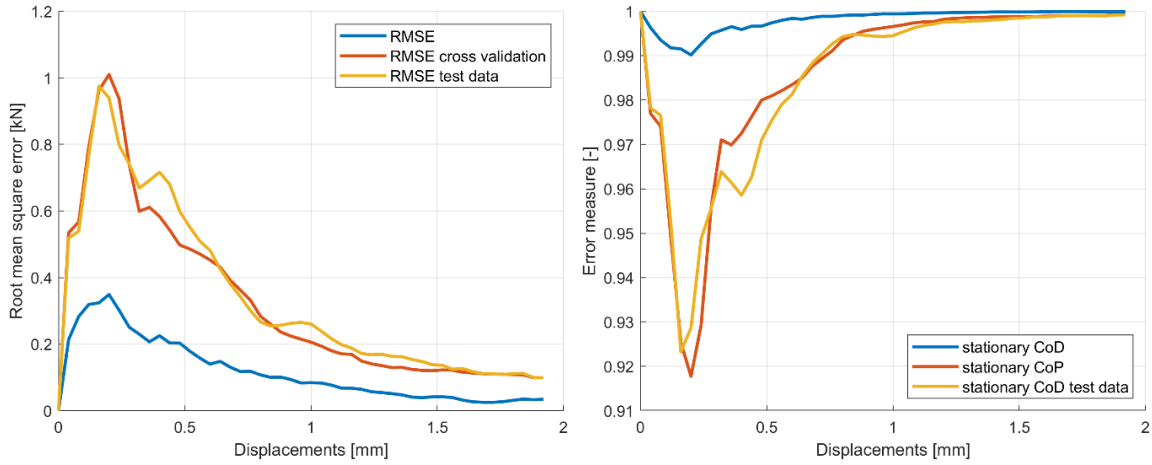

Figure 21. Wedge splitting test example: RMSE of the model fit from the cross-validation residuals and from the test samples (left) and the corresponding stationary CoD and CoP values (right)

The estimated RMSE errors of the 200 training samples are shown for each displacement value in Figure 21 together with the cross-validation results and the RMSE estimates from an independent test data set with 1000 Latin Hypercube samples. The figure indicates, that the cross-validation estimates agree very well with the test samples. Since the standard deviation of the response decreases significantly with increasing displacements as shown in Figure 20, we consider the stationary CoD and CoP according to Equation 22 and 24. In Figure 21 both measures are shown together with the stationary CoD of the test data. As expected, all three quantities will approach to one for increasing displacements since the RMSE estimates decrease. If the ordinary CoD and CoP would be used instead of the stationary measures, the predicted approximation quality would decrease for larger displacements due to the reduced variation of the displacements in the samples.

4 Conclusions

In this paper, statistical measures for the assessment of the prediction quality of machine learning models are investigated regarding their accuracy and robustness. Based on a cross-validation approach, the Coefficient of Prognosis was introduced as a model independent quality measure. However, the implementation of the cross-validation procedure is very important for a stable estimation of the prediction quality as shown in the numerical examples. From these findings, we would prefer the k-fold cross-validation towards the Leave-one-out approach since it gives more conservative estimates especially for a limited number of training data points.

Statistical confidence bounds of these global quality measures have been derived by using the bootstrap approach, whereas the resampling was evaluated directly on the cross-validation residuals. Therefore, this procedure can be applied without any additional model training. By means of several numerical examples, the value of the estimated confidence bounds could be demonstrated. This additional information helps to decide, how reliable the quality estimators are, if further data points are necessary, or if the prediction quality is affected by possible outliers.

Additionally, to the global quality measures, we introduced the local Root Mean Squared Error (RMSE) and the local CoP as local quality measures, which can be evaluated for each approximation point. They offer model independent error estimators of the local model prediction, which could be very valuable for Digital Twins applications.

The extension to non-scalar outputs requires the unique mapping of the discretization to a reference mesh, where the prediction error of each discretization point could be evaluated similarly to a scalar output. In order to obtain a unitless measure as the CoD and CoP, a normalization could be realized using the individual variation of each discretization point or assuming a stationary output variation. Both procedures require an efficient implementation as realized in Statistics on Structures [37], [45] as part of the Ansys optiSLang software package [32].

5 Acknowledgement

This article is dedicated to Prof. Christian G. Bucher, former professor at the Bauhaus-University in Weimar, Germany, and Technical University in Vienna, Austria. Prof. Bucher supported the work of the former Dynardo GmbH over more than 20 years with his excellent expertise and knowledge.

Additionally, we want to thank Dr. Johannes Will, who is the founder of the former Dynardo GmbH, for his huge commitment to the success of the Metamodel of Optimal Prognosis approach within the Ansys optiSLang community.

6 References

[1] Myers, R. H., and Montgomery, D. C., 2002, “Response surface methodology: process and product optimization using designed experiments,” John Wiley & Sons.

[2] Montgomery, D. C., and Runger, G. C., 2003, “Applied Statistics and Probability for Engineers,” third ed. John Wiley & Sons.

[3] Krige, D., 1951, “A statistical approach to some basic mine valuation problems on the Witwatersrand,” Journal of the Chemical, Metallurgical and Mining Society of South Africa, 52, pp. 119–139.

[4] Lancaster, P., and Salkauskas, K., 1981, “Surface generated by moving least squares methods,” Mathematics of Computation, 37, pp. 141–158.

[5] Park, J., and Sandberg, I., 1993, “Approximation and radial basis function networks,” Neural Computation, 5(2), pp. 305–316.

[6] Smola, A., and Schölkopf, B., 2004, “A tutorial on support vector regression,” Statistics and computing, 14(3), pp. 199–222.

[7] Hagan, M. T., Demuth, H. B., and Beale, M., 1996, “Neural Network Design,” PWS Publishing Company.

[8] Goodfellow, I., Bengio, Y., and Courville, A., 2016, “Deep learning,” MIT press.

[9] Herrmann, L., and Kollmannsberger, S., 2024, “Deep learning in computational mechanics: a review,” Computational Mechanics, pp. 1–51.

[10] Ye, P., 2019, “A review on surrogate-based global optimization methods for computationally expensive functions,” Software Engineering, 7(4), pp. 68–84.

[11] Cheng, K., Lu, Z., Ling, C., and Zhou, S., 2020, “Surrogate-assisted global sensitivity analysis: an overview,” Structural and Multidisciplinary Optimization, 61, pp. 1187–1213.

[12] Bucher, C., and Most, T., 2008, “A comparison of approximate response functions in structural reliability analysis,” Probabilistic engineering mechanics, 23(2-3), pp. 154–163.

[13] Moustapha, M., Marelli, S., and Sudret, B., 2022, “Active learning for structural reliability: Survey, general framework and benchmark,” Structural Safety, 96, p. 102174.

[14] Yondo, R., Bobrowski, K., Andres, E., and Valero, E., 2019, “A review of surrogate modeling techniques for aerodynamic analysis and optimization: current limitations and future challenges in industry,” Advances in evolutionary and deterministic methods for design, optimization and control in engineering and sciences, pp. 19–33.

[15] Westermann, P., and Evins, R., 2019, “Surrogate modelling for sustainable building design–a review,” Energy and Buildings, 198, pp. 170–186.

[16] Zhang, W., Gu, X., Hong, L., Han, L., and Wang, L., 2023, “Comprehensive review of machine learning in geotechnical reliability analysis: Algorithms, applications and further challenges,” Applied Soft Computing, 136, p. 110066.

[17] Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K., 2005, “Surrogate-based analysis and optimization,” Progress in aerospace sciences, 41(1), pp. 1–28.

[18] Forrester, A., Sobester, A., and Keane, A., 2008, “Engineering design via surrogate modelling: a practical guide,” John Wiley & Sons.

[19] Browne, M. W., 2000, “Cross-validation methods,” Journal of mathematical psychology, 44(1), pp. 108–132.

[20] Kleijnen, J. P., and Sargent, R. G., 2000, “A methodology for fitting and validating metamodels in simulation,” European Journal of Operational Research, 120(1), pp. 14–29.

[21] Molinaro, A. M., Simon, R., and Pfeiffer, R. M., 2005, “Prediction error estimation: a comparison of resampling methods,” Bioinformatics, 21(15), pp. 3301–3307.

[22] Bischl, B., Mersmann, O., Trautmann, H., and Weihs, C., 2012, “Resampling methods for meta-model validation with recommendations for evolutionary computation,” Evolutionary Computation, 20(2), pp. 249–275.

[23] Beck, J. L., and Yuen, K.-V., 2004, “Model selection using response measurements: Bayesian probabilistic approach,” Journal of Engineering Mechanics, 130(2), pp. 192–203.

[24] Park, I., Amarchinta, H. K., and Grandhi, R. V., 2010, “A bayesian approach for quantification of model uncertainty,” Reliability Engineering & System Safety, 95(7), pp. 777–785.

[25] Most, T., 2011, “Assessment of structural simulation models by estimating uncertainties due to model selection and model simplification,” Computers and Structures, 89, pp. 1664–1672.

[26] Bucher, C., 2018, “Metamodels of optimal quality for stochastic structural optimization,” Probabilistic Engineering Mechanics, 54, pp. 131–137.

[27] Niehoff, M., Bestle, D., Kupijai, P., et al., 2024, “Model-based design optimization taking into account design viability via classification,” Engineering Modelling, Analysis and Simulation, 1(1), pp. 1–12.

[28] Escribano, N., Bielsa, J. M., and Lahuerta, F., 2024, “pymetamodels: A python package for metamodeling and design automation,” SoftwareX, 26, p. 101735.

[29] Most, T., and Will, J., 2011, “Sensitivity analysis using the Metamodel of Optimal Prognosis,” In 8th Optimization and Stochastic Days, Weimar, Germany, 24-25 November, 2011.

[30] Most, T., Gräning, L., Will, J., and Abdulhkim, A., 2022, “Automatized machine learning approach for industrial application,” In NAFEMS DACH conference, Bamberg, Germany, 4-6 October 2022.

[31] Kvålseth, T. O., 1985, “Cautionary note about R2,” The American Statistician, 39(4), pp. 279–285.

[32] Ansys Inc., 2023, “optiSLang documentation: methods for multi-disciplinary optimization and robustness analysis”.

[33] Abdulhkim, A., Gräning, L., and Most, T., 2022, “Automated design of architectures of artificial neural networks,” US Patent application 63/211,185.

[34] Jones, D. R., Schonlau, M., and Welch, W. J., 1998, “Efficient global optimization of expensive black-box functions,” Journal of Global optimization, 13(4), p. 455-492.

[35] Efron, B., 1992, “Bootstrap methods: another look at the jackknife,” In Breakthroughs in statistics: Methodology and distribution. Springer, pp. 569–593.

[36] Most, T. and Knabe, T., 2010, “Reliability analysis of the bearing failure problem considering uncertain stochastic parameters,” Computers and Geotechnics, 37(3), pp. 299–310.

[37] Wolff, S., 2016, “Random fields and field meta models – correlation analysis in time and space,” Dynardo RDO Journal, 2016(1), pp. 2–8.

[38] Bayer, V., Kunath, S., Niemeier, R., and Horwege, J., 2018, “Signal-based metamodels for predictive reliability analysis and virtual testing,” Advances in Science, Technology and Engineering Systems Journal, 3(1), pp. 342–347.

[39] Cremanns, K., 2021, “Probabilistic machine learning for pattern recognition and design exploration,” Phd thesis, RWTH Aachen, Germany.

[40] Most, T., Gräning, L., Wolff, S., and Cremanns, K., 2024, “Automatisierte Approximation von CAE-Signal- und Feldergebnisgrößen mit Methoden des Maschinellen Lernens,” NAFEMS DACH Magazin, 70, pp. 32–40.

[41] Huntington, D., and Lyrintzis, C., 1998, “Improvements to and limitations of Latin hypercube sampling,” Probabilistic engineering mechanics, 13(4), pp. 245–253.

[42] Stander, N., Basudhar, A., Gandikota, I., Liebold, K., Svedin, A., and Keisser, C., 2021, “LS-OPT status update,” In Proc. 13th European LS-DYNA Conference, Ulm, Germany.

[43] Most, T., Rasch, M., Ubben, P. T., Niemeier, R., and Bayer, V., 2023, „A Multimodal Importance Sampling Approach for the Probabilistic Safety Assessment of Automated Driver Assistance Systems,” Journal of Autonomous Vehicles and Systems, Vol. 3, 011001-1

[44] Most, T., 2005, “Stochastic crack growth simulation in reinforced concrete structures by means of coupled finite element and meshless methods,” PhD thesis, Bauhaus-Universität Weimar, Germany.

[45] Ansys Inc., 2021, Statistics on Structures User’s Guide.