EVALUATION APPROACH FOR MACHINE LEARNING CONCEPTS IN OCCUPANT PROTECTION BASED ON MULTI-ATTRIBUTE DECISION MAKING

Franz Plaschkies
Technische Hochschule Ingolstadt
Germany
Ketlen Possoli
Federal University of Santa Catarina; Technische Hochschule Ingolstadt
Brazil; Germany
Ondrej Vaculin
Technische Hochschule Ingolstadt
Germany
Axel Schumacher
University of Wuppertal
Germany
Pedro de Andrade Junior
Federal University of Santa Catarina
Brazil

Paper Number 23-0055

ABSTRACT

The systems for occupant protection in passive vehicle safety are primarily developed with single statistical representations of humans, so-called Anthropomorphic Test Devices (ATDs). Unfortunately, those ATDs cover additional features like age and body shape insufficiently during development. Augmenting finite element simulations with a metamodel trained by machine learning is promising to overcome this barrier. However, the database design, the machine learning architecture, and the requirements for quality and robustness influence each other. Therefore, objective criteria must be defined to compare the alternatives taking cost and benefit aspects under changing preferences into account. Having complex criteria can be framed as a multi-attribute decision-making problem. This paper's objective is the development of a transparent assessment scheme for virtual statistical simulation for rapid vehicle occupant safety assessment using supervised learning.
PROMETHEE is selected as an appropriate decision-making approach. A process, consisting of a sequential definition of the criteria leading to the final assessment, is proposed to adapt the method in this paper's domain. The methodology is tested on sample alternatives, generated using a calibration-type machine learning architecture and data from finite element simulations. The original PROMETHEE algorithm cannot handle a vast number of alternatives. Since, typically, numerous alternatives occur during the development of a machine learning application, a sorting-based modification is implemented.
Finally, the findings are discussed, and recommendations for related use cases are given. The proposed method seems applicable to the described domain and near-related ones. Moreover, multiple tendencies between an alternative's parameters and rank can be identified in the test samples.

INTRODUCTION

In the recent years, passive vehicle safety has been dominated by the increasing virtualisation of assessment methods. Historically, crash tests are performed with real prototypes. A single virtual, physical simulation utilising, e. g. Finite Element Analysis (FEA), comes with significantly lower cost, higher flexibility, and an unmatched insight into the physical processes. However, the virtual simulations must fit the reality sufficiently, which makes extensive validation necessary. The degree of model detail and computational effort has been increased to fulfil the demand for trustworthy models. Nowadays, an industrial simulation on state-of-the-art hardware takes hours to days. Multiple developments led to the need for further acceleration of virtual methods: (i) shorter product cycles require rapid assessment; (ii) increased parameter spaces make more efficient methods for a sufficient assessment necessary; (iii) Euro NCAP recently proposed in [1] scenario-based virtual testing; (iv) the development in autonomous driving will introduce a broad range of allowed sitting positions and activities during driving, as stated by Östling et al. in [2]; (v) the population of vehicle occupants is significantly more diverse than it was when the anthropometrics for the state-of-the-art crash test dummies were developed, as concluded by Reed et al. in [3] and Wang et al. in [4]. Those dummies, so-called anthropomorphic test devices (ATDs), are technical measuring devices. They are the 5^{th}, 50^{th}, and 95^{th} percentile representations of the North American population in the 1970s [5].

This paper proposes a method to develop a transparent assessment scheme for virtual statistical simulation for rapid vehicle occupant safety assessment using supervised learning. The methodology was tested on vehicle occupant safety assessment, specifically on the front crash case for passengers. The data originated from a simplified 2D FEA-model. The machine learning architecture contained a calibration approach introduced by Plaschkies et al. in [6] with supervised learning techniques.
Next to other influences, the above-described development sparked various publications of machine learning applications in the passive safety assessment, as summarised by Plaschkies et al. in [5]. The identified studies focused on prediction quality metrics like accuracy to assess and compare their investigated approaches. However, machine learning notably depends on the amount and quality of data leading to complex metrics with interacting parameters. Therefore, the trade-off between data generation costs and their value for the method must be represented. Approaches from Multi-Criteria Decision Making (MCDM) can provide a transparent way to compare different alternatives regarding multiple criteria to solve a particular problem.

STATE-OF-THE-ART BASED SELECTION OF THE DECISION-MAKING METHOD

Decision-making is a centuries-old problem; many publications have been dedicated to this topic. Hence, some assumptions must be declared before entering the state-of-the-art. Moreover, the problem described above implies a discrete nature of the alternatives. Furthermore, presumably, some criteria can be only described on an ordinal scale like a grading system. Finally, the purpose is to select the best alternatives from a given set, or to check, if a new alternative is beneficial. The complex situation will probably lead to numerous alternatives.
According to Hwang et al. in [7], the application of MCDM is widespread. However, there are some common characteristics between them: (i) incommensurable units, (ii) conflict between criteria, (iii) multiple objectives/attributes, and (iv) design/selection.
Some authors have divided MCDM into two categories. First, Multi-Attribute Decision-Making (MADM) focuses on problems with discrete decision spaces. Second, Multi-Objective Decision-Making (MODM) problems involve several competing objectives that need to be optimised simultaneously [8].
An MODM problem is associated with the problem of designing optimal solutions through mathematical programming. The number of possible decision alternatives can be immense. Usually, the decision space is continuous [9]. As common characteristics, MODM methods have: (i) a set of quantifiable objectives, (ii) a set of well-defined constraints, and (iii) a process of obtaining some trade-off information between quantifiable objectives and non-quantifiable objectives [7].
MADM requires that the choice is being made with clearly defined criteria. MADM problems have predetermined and limited number of alternatives; hence the decision space is discrete. Solving a MADM problem requires ordering and ranking [9, 10].
Comparing MODM and MADM, MADM seem to suit better the peculiarities of this paper's problem. The evaluation within a discrete decision space with predefined alternatives and criteria fits the declared assumptions. The number of alternatives is finite, although large.
Majdi divides in [11] MADM into four groups: Cost-Benefit Analysis (CBA), Elementary, Multi-Attribute Utility Theory (MAUT), and Outranking. CBA evaluates on a monetary basis the costs and benefits of the alternatives. Elementary methods do not need computation support and can be used with a few alternatives and criteria with a single decision-maker [12]. Examples of elementary methods are the Pros and Cons Analysis, the Maximin, and the Maximax Methods [11].
For the MAUT methods, Winterfeldt et al. described in [11, 13 apud] the procedure as: (i) evaluate alternatives, (ii) assign weights, (iii) aggregate the weights of attributes and alternative scores, and (iv) perform sensitivity analyses and make recommendations. For example, the Analytic Hierarchy Process (AHP) is a widely used method in this class. Advantages are the possibility to use qualitative and quantitative criteria and good traceability [14].
Outranking methods require specifying alternatives, criteria, and the use of data from the decision table. For example, the ELECTRE family (ELimination Et Choix Traduisant la REalite) consists of seven different models derived from the original one. The result is the smallest set of the best alternatives while providing no ranking with such a set [14].
The PROMETHEE approach (Preference Ranking Organisation Method for Enrichment Evaluations), described by Brans et al. in [15], is another outranking method based on extensions of the notion of criteria and can be relatively rapidly built by the decision maker. There are two base possibilities to provide rankings in this method: PROMETHEE I provide a partial pre-order, and PROMETHEE II the total pre-order. As per de Almeida et al. in [16], the method was for example extended for a range assessment in PROMETHEE III and the application on continuous decision spaces in PROMETHEE IV. According to Brans et al., PROMETHEE II is easier to handle by the decision maker. However, PROMETHEE I contain more realistic information, especially regarding incompatibilities [17].

PROMETHEE I considers the intersection between the positive and negative flows in a partial pre-order between the alternatives. The ranking of this partial pre-order can be represented as a network graph and contains information on the comparability of two alternatives. Non-comparability equals a not confirmed outrank. The combination of in- and out-flow determines if one alternative is outranking another or is indifferent.
PROMETHEE II classifies the alternatives, establishing a complete pre-order among all the alternatives using the net-flow. The alternatives with the higher net-flow are preferred over the ones with a lower net-flow.
Disadvantage of this method is that it is hard to keep an overview of the problem when many criteria are involved, and it can be time-consuming [14]. Despite those drawbacks, PROMETHEE was selected since it is widely used, does not require normalisation, and is applicable even when information is missing. Furthermore, there are many methods to assist in choosing the best option from a set of alternatives based on multiple criteria. However, it can be challenging to assess which method is the most appropriate to use in each situation or even which questions to ask when comparing various methods [18].
The original PROMETHEE algorithm described by Brans et al. in [15] is displayed on the left side of Table 1. Equation (1) represents the preference π for the criterion k of the alternative a_i over another alternative x. For the sake of simplicity a usual preference function was used here to evaluate the criteria value f. Each criterion has a weight w assigned by the decision maker. Again, for simplicity, an equal weight for all q criteria were chosen. In the first step, the preference of one alternative over all other alternatives is calculated according to equation (3), where n is the total number of alternatives A. Next, the PROMETHEE I in-flow ϕ^+ by equation (4) and out-flow ϕ^- by equation (6) is determined. Finally, the PROMETHEE II total pre-order in form of the net-flow ϕ is derived in equation (9).
Analysing the algorithm, the time complexity is $\mathcal{O}(qn^2)$. PROMETHEE is a fully deterministic procedure; the same input will lead to the same output. However, there are some instabilities regarding the pre-order; also described, e. g. by de Keyser et al. in [19], as the reverse rank problem. While in the direct comparison of two alternatives, the preference matrix remains the same, the flow calculation introduces a dependency of the pre-order on the compared alternatives. The reverse rank problem requires the re-assessment of all alternatives if a new one is added. Revisiting the above-declared assumptions and the time complexity, PROMETHEE seems to face a significant hurdle.
Calders et al. proposed in [20] an adaption of the original algorithm achieving a time complexity of $\mathcal{O}(qn \log n)$. This massively reduced complexity enables the computation of huge numbers of alternatives. As shown on the right side of Table 1, the uni-criterion flows are calculated according to equations (5), (7), and (8) as the initial step. Calders et al. observed that the values of a criterion per alternative can be sorted individually, allowing to infuse established and highly efficient sorting algorithms leading finally to reduced time complexity. Comparing equations (11) and (12), it becomes clear that for PROMETHEE II, the complete pre-order is for both methods the same. As a drawback, only the PROMETHEE II result can be obtained.

Table 1.
Comparison of original and sorting-based algorithms

$$\pi_k(a_i, x) = \begin{cases} 0 \text{ if } f_k(a_i) \geq f_k(x) \\ 1 \text{ if } f_k(a_i) < f_k(x) \end{cases} \quad (1)$$

$$w_k = {}^1/_q \quad (2)$$

Original PROMETHEE I & II [15]

$$\pi(a_i, x) = \sum_{k=1}^{q} [w_k * \pi_k(a_i, x)] \quad (3)$$

$$\phi^+(a_i) = \frac{1}{n-1} \sum_{x \in A} \pi(a_i, x) \quad (4)$$

$$\phi^-(a_i) = \frac{1}{n-1} \sum_{x \in A} \pi(x, a_i) \quad (6)$$

$$\phi(a_i) = \phi^+(a_i) - \phi^-(a_i) \quad (9)$$

$$\phi(a_i) = \frac{1}{q(n-1)} \sum_{x \in A} \left[\sum_{k=1}^{q} [\pi_k(a_i, x) - \pi_k(x, a_i)] \right] \quad (11)$$

Sorting Based PROMETHEE II [20]

$$\phi_k^+(a_i) = \frac{1}{n-1} \sum_{x \in A} \pi_k(a_i, x) \quad (5)$$

$$\phi_k^-(a_i) = \frac{1}{n-1} \sum_{x \in A} \pi_k(x, a_i) \quad (7)$$

$$\phi_k(a_i) = \phi_k^+(a_i) - \phi_k^-(a_i) \quad (8)$$

$$\phi(a_i) = \sum_{k=1}^{q} [w_k * \phi_k(a_i)] \quad (10)$$

$$\phi(a_i) = \frac{1}{q(n-1)} \sum_{k=1}^{q} \left[\sum_{x \in A} [\pi_k(a_i, x) - \pi_k(x, a_i)] \right] \quad (12)$$

PROPOSED METHOD

In this paper a stepwise method to the MCDM problem is proposed. As displayed in Figure 1, the approach consists of the definition of an initial criteria list, the derivation of a final criteria list, and the decision making.

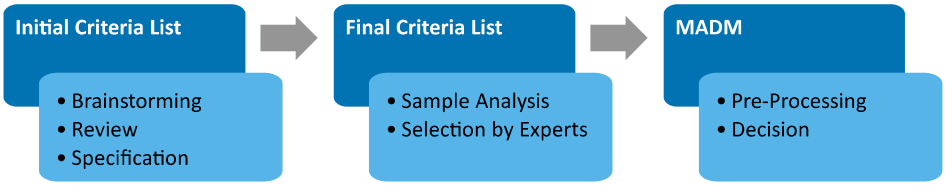

Figure 1. Proposed method flow

Initial Criteria List

Brainstorming phase To define criteria, a brainstorming session with experts is proposed. Categories of cost and use factors can support collecting the criteria.

The central use of a metamodel is determined by its estimation quality. This quality should not only be defined by typical metrics such as accuracy or recall but also consider the detail of degree and the relevance of an estimation. Furthermore, the difference in result generation between the assessed alternative and the simple FEA simulation can be considered.

The main cost factor is induced by the data for training and assessment of the metamodel. If the architecture requires additional data as input for each estimation, it adds to the costs. Depending on the data volume, the computational cost for the training and assessment cycle and even per prediction can be relevant.

The end of a model's validity should be considered. In this case, additional costs through data generation for retraining the metamodel will occur. Furthermore, it is assumed that over time and continuous development, the vehicle deviates increasingly from the ones used for metamodel training. Hence, a later loss of validity – or a wider validity range – would mean a higher model value.

Review & specification phase Typically, brainstorming techniques are suitable for collecting ideas efficiently; however, completeness is not guaranteed. Hence, a review checking the inner logic and completeness of the criteria is recommendable. Dropping criteria in this step is unnecessary; this will be done in the final specification phase.

During the review phase, the reporting scale and assessment method should be defined for each criterion. The reporting scale will influence the selection of a suitable MADM method. The assessment method's exact definition will help to review the selected criteria and is the prerequisite for the later steps. Since criteria for the actual assessment should be selected later, the documentation of each criterion and its motivation is necessary.

Derivation of the Final Criteria List

Ideally, the list of criteria from the above steps can assess all relevant aspects of possible alternatives. However, highly correlated criteria are likely to occur since the described approach prefers adding criteria over dropping them. Therefore, the authors propose to create multiple samples of alternatives. Those should be used to test the validity and plausibility of the defined assessment algorithms and for another review phase. The samples can support the identification of highly correlated criteria. Those criteria would potentially assess the same aspect; hence assign a higher weight to such an aspect.

It must be noted that the sample alternatives will not cover all possible cases. Henceforth, expert opinion is needed to interpret the findings correctly. Each criterion and the related findings should be discussed considering the aspects described in Table 2.

Table 2.
Aspects to consider during criterion-selection

Representativeness
The representativeness of the generated samples determines if the criteria are correct and meaningful and represent diverse aspects of the problem.
Correlation
Correlated criteria should be merged to avoid unwanted higher weights on a specific aspect. Invariant criteria should be inspected if the invariance is only due to the generated samples or meaningful for the overall problem. In the first case, the criterion can remain, in the ladder, dropped.
Transparency & Directness
The criteria should be grouped into meaningful categories to support a transparent rating scheme. It depends on the actual use case to which category an aggregated criterium fits. Another aspect regarding transparency is the understandability of a criterium. A directly assessed criterium is more straightforward to process and understand than one resulting from complex calculations.

Level of the scales
Ultimately, the reporting scale of a criterion should be considered. Typically, nominal-, ordinal-, interval-, and ratio-scales are used, where nominal has the lowest and ratio the highest level. The lowest-level scale of all used ones will determine which MCDM method can be utilised. Not all criteria can be assessed on ratio scales. However, if possible, the higher-level scale always seems preferable.

Multi-Attribute Decision Making

Following the state-of-the-art analysis, to solve the decision-making problem, the PROMETHEE II method was selected. The adaption of Calders et al. in [20] seems recommendable to deal with the expected high number of alternatives despite the loss of the PROMETHEE I result.

The method required selecting a preference function; it is influenced by the lowest scale order and the user's taste. If of all criteria, the lowest order scale is ordinal, only the usual-criterion can be used. Other definitions, like the linear- or step-criterion, require proportional intervals between the variables.

APPLICATION

Computations were executed on a workstation equipped with an Intel Xeon W-2123 CPU with 3.6 GHz and 64 GB RAM. The cluster used for the larger FE-simulations had per node two Intel Xeon E5-2687W v4 CPUs with 3 GHz.

All described algorithms were implemented in Python 3.8. The neural network used for machine learning was taken from the Scikit-Learn library [21] version 1.0.1. Database-related operations like sorting were done utilising the Pandas library [22] version 1.4.2. All FE-simulations were performed in LS-Dyna 10.0 (MPP on cluster, SMP on workstation) with single precision.

Database

FE-model To test the method assessment approach, a database from a recent study [6] was used. The simulations were done with a 2D rigid body model, as shown in Figure 2, representing an occupant undergoing a frontal crash. Five anthropometrical configurations were created, orienting on the common crash test dummies with the 5th, 50th, and 95th percentiles. The 25th and 75th percentiles were added by interpolation. A Full Factorial Design of Experiment (DoE) was defined, containing the variation of backrest angle, seat ramp angle, impact speed, and the force of the shoulder belt load limiter. Each factor was varied in six levels, and the resulting DOE of size 1,296 was repeated for the five occupant sizes leading to a total of 6,480 simulations.

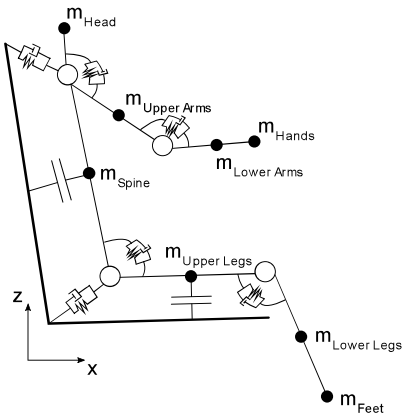

Figure 2. Occupant model 2D

The model has not been validated; thus, the physical behaviour seems overall plausible. Few simulations suffered numerical instabilities and were dropped as outliers. Following the recent study, the maximum resultant chest acceleration lasting at least 3 ms $a_{Chest,a3ms}$ and the maximum head forward displacement relative to the vehicle $x_{Head,Local}$ was selected for the investigations. The selection was motivated by stable numerical outputs, the model's capabilities, and biomechanical relevance.

For the seat, a motion was prescribed, taking the pulse generated by FE-simulation with a Toyota Yaris 2010 model [23]. The load cases were defined as vehicles crashing frontally into a rigid barrier with different velocities. One crash simulation took approximately three hours using a single node on the cluster. In comparison, one occupant simulation on the workstation accounted for approximately three minutes.

Machine learning architecture For the machine learning, in a previous study [6] an architecture was sketched as depicted in Figure 3. The key characteristic of this architecture is the hybrid approach of providing a calibration simulation for each prediction. The calibration is a physical simulation of an anthropometrical reference configuration. The predicted outputs of the metamodel are selected results of different anthropometrical

configurations, but in the same vehicle environment as the reference. As learning algorithm, a deep neural network with two hidden layers was selected.

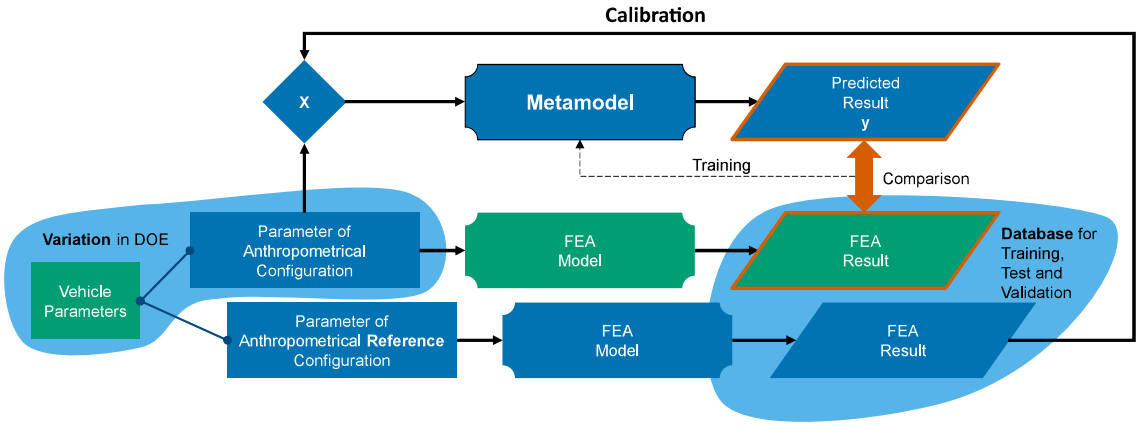

Figure 3. Machine learning architecture in [6]

For later assessment, a characterisation of the environment must be defined. Vehicle parameters are not explicitly given but implicitly contained in the calibration simulation. The advantage is that changes in vehicles that are different than expected or unquantifiable can be covered. The drawback, however, is an unclear defined field of variation in which the metamodel was trained, the so-called validity field. Hence, the transition between interpolation and extrapolation cannot be derived directly. However, the calibration simulation contains information on the environment since a unique setting will result in a unique response.

In the case of the used 2D model, the occupant's behaviour can be described by the kinematics of the joint and endpoints (head, shoulder, elbow, hand, hip, knee, foot). The relative displacement to the vehicle of those points and the global acceleration provides sufficient insight. To measure the position of an environment relative to the validity field in a transparent manner, a reduction to one or two dimensions seems necessary.

Principal Component Analysis (PCA) was selected, and its Scikit-Learn implementation was used. This linear and self-centring method derives from high dimensional data principal components by eigenvalue decomposition, explaining the variance in the data. Such components do not necessarily correspond with single DOE parameters; they can be combinations of them, too.

Before applying PCA, the data had to be transformed. First, the sensor signals were smoothed using a CFC60 filter [24], and simulations suffering numerical instabilities were removed. Second, the sensor output time series were arranged line-wise. Each column was a discrete timestamp from a particular sensor as a dimension. Third, each line contains the data from one FE-simulation as samples. Last, each dimension/column was standardised by subtracting the dimension's mean value from each sample and then dividing it by the standard deviation.

Applied to the dataset of 50th percentiles, as displayed in Figure 4, the first principal component explains ca. 38 % of the variance and the second additionally ca. 14 %. A 6x6 field of distinct islands is observable. The first component could be associated with the six discrete impact speed settings, and the second one with the six discrete backrest angle settings from the DOE. Despite the relatively low explained variance of the principal components, only the first one was used for further processing since it could have been associated with the impact velocity and for simplicity in showcasing the actual decision-making method.

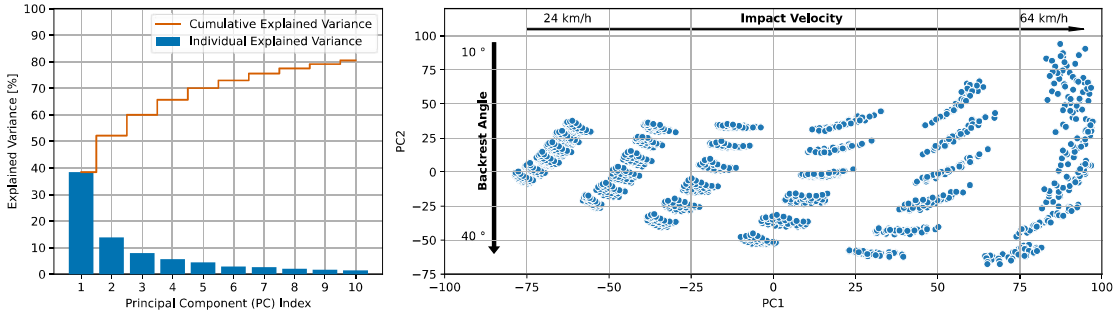

Figure 4. Result of PCA analysis

Due to the dimensional reduction, it was possible to differentiate between interpolation and extrapolation. Therefore, data from the interpolation field was used for training and testing the metamodel. The data from the extrapolation field was used for validation.

Generated alternatives In combination with the given database of physical models, the selected architecture allows to investigate numerous alternatives to create a metamodel. First, the calibration was varied between the 50th percentile and the other edge percentiles. Second, the calibration contained $a_{Chest,a3ms}$, $x_{Head,Local}$, or both. Third, the same variation was used for the predictions. Those labels can be defined in two or more classes or as continuous values. The number of predictable percentiles is directly dependent on the used calibrators. A percentile used as a calibrator cannot be utilised in the predictions. Finally, the hyperparameters of the neural network (number of layers, number of neurons per layer) were varied. The first component of the PCA on the 50th percentile data was selected characterise to the environment. Through this, the interpolation field could be varied as well as the number of simulations in it. In total, 12,960 alternatives were generated.

Initial List of Criteria

Brainstorming phase Following the method described above, the three categories, metamodel-setup-cost, usage, and validity-range, were defined in the first step. Those categories represent the live cycle cost and use factors. Next, a group of experts filled the categories with relevant criteria in a brainstorming session.

Criteria in the metamodel-setup-cost category should assess all occurring costs associated with creating a metamodel. Therefore, the category was differentiated into the cost for the physical simulation database, the training costs, the testing, and the assessment of the validity range.

The usage category focuses on the live cycle phase in which the metamodel is utilised. In this phase, costs for each prediction exist, but the value of the predictions is also shown.

For the last category, the validity range, it is assumed that at one point, the vehicle under development deviates so much from the ones used for the metamodel setup that its validity is compromised. In this case, costs for retraining or tuning will occur.

Review & specification phase After the brainstorming session, the experts reviewed and restructured the criteria and defined their reporting scales. The selected criteria are discussed below and are listed in Table 4 of the appendix. As documented in the table, ultimately, not all criteria were found to be implementable.

As metamodel-setup-costs, criteria assessing computation time and the number of simulations or samples were accounted. It was differentiated between computation time for the crash and occupant simulations (~ 3 h, ~ 3 min). Computation time reports on a continuous scale whereas the sample number on a discrete scale. For both, lower is seen as better.

In the usage section, the use and value of a metamodel were locked from several angles. First is the value from the prediction type; a binary classification is seen to have a lower value than a continuous regression. A rating system was used as a metric. Second, a single sensor's output detail can determine the value. With decreasing value, the prediction of the entire sensor output as time series, the prediction of relevant output characteristics, and finally, the prediction of a single value was defined in a rating system. Third, a crash test dummy is instrumented with numerous sensors. More the sensors are used, higher the value. This criterion was defined as the number of not used sensors to fit into the lower-is-better scheme. Of course, not all sensors have the same relevance. The defined criterion reports by relevant legislation, consumer ratings, and physics in a ranking scale. Finally, the granularity of the predicted anthropometrical configuration can range from a single configuration over distinct percentiles up to the variation of anthropometrical measures.

The most apparent and commonly used criterion is the prediction quality metric. For regression, the coefficient of determination R^2 was used. For classification cases, the F-score was selected. Both metrics report to a continuous scale where one is the best. The R^2-score can take negative values; to adapt its scale to F-score, equation (13) was defined.

$$R^2 = \begin{cases} 0 \text{ if } R^2 < 0 \\ R^2 \text{ if } R^2 \geq 0 \end{cases} \tag{13}$$

As described above, the environment was characterised as 1D through PCA. As the value of a metamodel increases, a new environment can differ from more the training field without compromising the model's validity. The machine learning metric was evaluated, as displayed in Figure 5, for the inter- and extrapolation zones separately to assess the width. Each zone was split into three segments to get a gradual result. It must be noted that a machine learning metric is a statistical measure and hence, needs an appropriate sample size to deliver a valid assessment [25]. Finally, the width results from the area in which the machine learning metric is continuously higher than 0.8. The assessed machine learning score was defined as the mean value of the machine learning metric over that width. Again, the machine learning metric was subtracted from one to achieve the lower-is-better scale, and the width was multiplied by minus one.

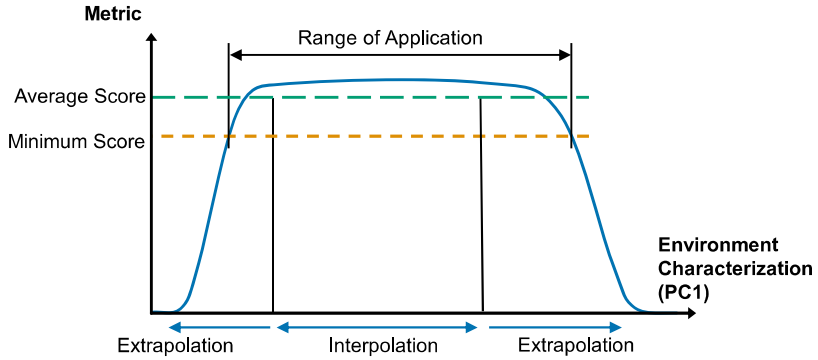

Figure 5. Concept of interpolation, extrapolation, and validity range

Final List of Criteria

To review the listed criteria and filter them, they were assessed in the 12,960 sample alternatives. For statistical insight, the Kendall correlation coefficient was used. This coefficient considers the rank correlation between two criteria and can deal with limited non-linearity and outliers.

The purposes of the correlation analysis were to support (i) the identification of double-assessed properties and (ii) the reasonability of the assessments. A few mistakes and misalignments in the assessment algorithms could be identified during this process. Furthermore, some criteria were found to be correlated or invariant.

Invariance of criteria occurred since the sample alternatives did not cover all possibilities identified by the experts. Such criteria were kept. Highly correlated criteria were merged.

Criteria related to computation time were under discussion. The values used for the computation time of the FE-models for crash and occupant simulation were average values; hence reliable. In contrast, the times from the instrumented assessment codes were measured only once. Hence, disturbances on the CPU, e. g. other processes, can lead to incomparable times for computation. However, for this paper, it was possible to run the process on a CPU exclusively. A statistical univariant analysis showed no extreme outliers; an example is shown in Figure 6. On this base, it was decided to keep those criteria since no adequate alternative could be identified.

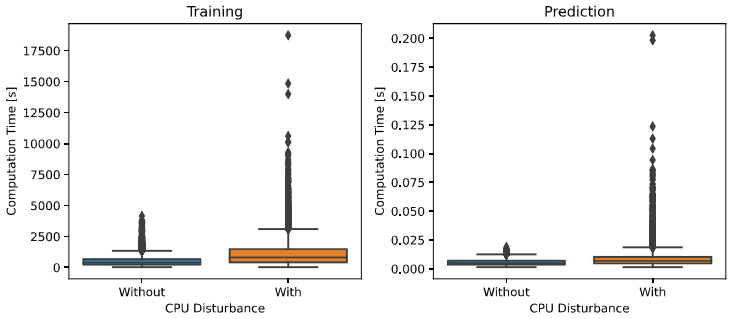

Figure 6. Computation time distribution with and without disturbances

Ultimately, 28 criteria were selected for further usage. During the selection process, 15 criteria were dropped. Also, the not implementable ones were removed. The final list is provided in Table 5 of the appendix.

Multi-Attribute Decision Making

Sorting method The alternation of the original PROMETHEE algorithm proposed by Calders et al. was implemented. The sorting-based method was described for a maximise-problem and a linear preference criterion. In comparison, the here used implementation inverted the comparing algorithm to a minimise-problem. The lowest order scale type in the final list of criteria was the ordinal scale. Hence, only the usual criterion could be used. The algorithm was adapted accordingly. For sorting, the MERGESORT algorithm was used. The final algorithm, as implemented, is displayed in Table 3.

The 12,960 sample alternatives with 28 criteria were divided into 40 chunks to check the algorithm and compare the computation time. Each chunk was assessed by implementations of the original PROMETHEE algorithm and the one with the Calders modification. Running on a workstation as a single CPU process, the median computation time of the original algorithm was 80.1 s and of the sorting method 0.1 s. The PROMETHEE II net-flows were within the limits of the computational precision same. The equality of both approaches and the drastically lower time complexity of the sorted approach was confirmed.

Table 3.
Algorithmic representation of PROMETHEE II implementation

Input: Number of alternatives n, Number of criteria q, criteria values assessed for all alternatives $f_{1\dots q}(a_{1\dots n})$ Return: Net-flows $\phi(a_{1\dots n})$		
1	SET w TO $1/q$	# Equal weight per criterion
2	INIT $\phi_{1\dots q}[a_{1\dots n}]$	# Uni-criterion net-flow
3	FOR k IN $f[a_0]$	
4	INIT $\phi_k^{+/-}[a_{1\dots n}]$	# Uni-criterion in- / out-flow
5	FOR $\mathbb{S}$ IN [1, -1]:	# In- & outflows
6	SET $\mathbb{f}_k(a)$ TO $f_k(a) * \mathbb{S}$	# Symmetry of flows
7	SET $\mathbb{f}_k(\mathbb{a})$ TO SORTED_DESCENDING $\mathbb{f}_k(a)$	# Merge Sort (a differs from $\mathbb{a}$ only in its order)
8	SET R TO $\mathbb{f}_k(\mathbb{a})$	# For the first object, all others are on the right
9	SET $\phi_k^{\mathbb{S}}[\mathbb{a}_1]$ TO 0	# The first alternative in order always has flow 0
10	FOR i IN $\mathbb{a}_{2\dots n}$:	# Loop over the following alternatives in order
11	SET $\phi_k^{\mathbb{S}}[\mathbb{a}_i]$ TO $\phi_k^{\mathbb{S}}[\mathbb{a}_{i-1}]$	# Start with the previous flow
12	WHILE $R[\mathbb{a}_1] > \mathbb{f}_k[\mathbb{a}_i]$	# Check for preference
13	DELETE $R[\mathbb{a}_1]$	# Move to left
14	SET $\phi_k^{\mathbb{S}}[\mathbb{a}_{i-1}]$ TO $\phi_k^{\mathbb{S}}[\mathbb{a}_{i-1}] + \frac{1}{n+1}$	# Add as the preference to uni-criterion in- / out-flow
15	FOR i IN $a_{1\dots n}$	
16	SET $\phi_k(a_i)$ TO $\phi_k^+(a_i) - \phi_k^-(a_i)$	# Uni-criterion net-flow
17	INIT $\phi(a_{1\dots n})$	# Net-flow
18	FOR i IN $a_{1\dots n}$	
19	SET $\phi(a_i)$ TO $\sum_{k=1}^{q}(w * \phi_k(a_i))$	# Net-flow

Result For the final assessment, all alternatives were tested as a whole and sorted by their PROMETHEE II complete pre-order. As defined above, seven parameters were varied: (i) the configuration of the neural network, (ii) prediction type, (iii) target percentile(s), (iv) calibrating percentile(s), (v) sample size, (vi) interpolation range, (vii) sensor(s) used in target(s), and (viii) sensor(s) used in feature(s). The tendencies, observed in Figure 7, are described below.

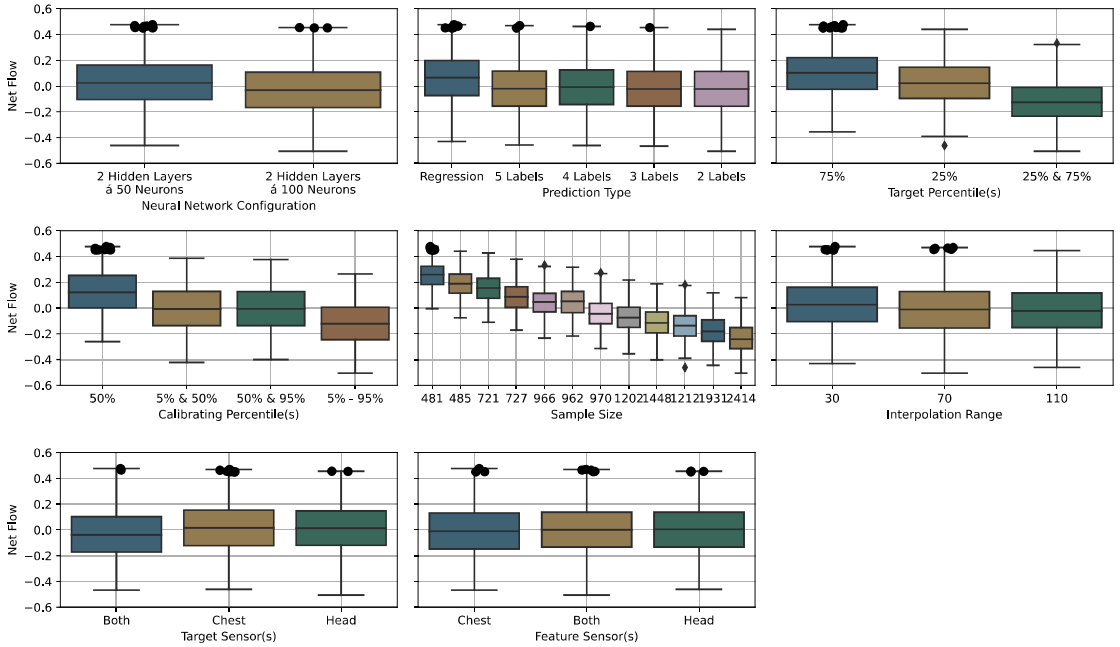

Figure 7. Net-flows evaluated for 12,459 alternatives – box plots with top 10 overlay

The results indicated a negative influence of sample size on the rank. One reason can be the increased cost of data generation, while other factors overlay the potential positive influence on the prediction quality.
Furthermore, the regression algorithms seemed slightly better than the others. The 95^{th} or 5^{th} percentile prediction seemed to be more successful than simultaneously targeting both. Predicting the 95^{th} percentile is indicated as beneficial. Using the 50^{th} percentile as a calibrator seems to be better than the 5^{th} or 95^{th} percentile. Taking the 5^{th} and 95^{th} percentile as calibrators does not seem advantageous. Finally, it seems that a tighter interpolation field has light benefits. The other varied parameters do not indicate preferences.
Concluding, the alternatives could be ranked using PROMETHEE II. The first analysed tendencies seem to be reasonable. In general, settings which compromise the prediction value have a strong influence.

DISCUSSION

The process of deriving the list of criteria seems, overall, a good concept. Nevertheless, the strong dependency of all steps on the knowledge and judgement of the involved experts must be pointed out.
The results of a brainstorming process can be unorganised, and there is no guarantee of completeness. Additionally, there is a chance for non-implementable criteria. The pre-declaration of some categories representing the main cost and use factors was very helpful. It is highly recommendable to invest already during that first phase in documenting each criterion's intentions. In the last review step, each criterion should be described extensively, helping to keep the overview and to succeed in the later steps. Concluding, with the proposed process, a comprehensive list can be created.
For the sake of simplicity, it seems recommendable to define all scales in a lower is a better manner. However, this is not required by algorithms as PROMETHEE. Furthermore, the ordinal scale can be applied to all criteria and can be assessed transparently. However, the choice of this scale limits the usable decision-making methods. Already using another preference function within PROMETHEE would transform the scale unwanted and unreasonably into a ratio scale [19].
As stated above, a set of test samples cannot represent all possible variations. The high number of alternatives used in this paper was mainly motivated to ensure a good range of variation and to enable the investigation of correlations. In the end, the correlations did not lead to a data-driven decision over the criteria. However, as a tool to detect unplausible behaviour, it was invaluable. It can be achieved with a significantly smaller number of test alternatives. By experts' judgement, a minimum number covering a maximal range of variations can be defined.
The results from the ranked alternatives originate in the 2D FE-model. A significant limitation can be found in the characterisation of the environment. First, the explained variance seemed insufficient even if the dimensional reduction showed a physically relatable result. In future studies, a detailed analysis on the base of a validated FE-model should be conducted and the method for dimensional reduction refined. Second, especially the criteria were defined for the narrow use case of supervised machine learning for the virtual assessment of occupant crash safety. If the method should be applied in deviating domains, each step starting with the declaration of the initial categories, should be reviewed. Depending on the complexity, increasing the number of test samples seems apt.
If changing the MADM method, the investigations on its behaviour and the parametric sensitivity should be done. Furthermore, especially if the exact rank of the assessed alternatives is relevant, the rank reversal issue of pairwise comparison-based methods, in general, but especially PROMETHEE II, should be assessed.

CONCLUSIONS AND OUTLOOK

The selection of an appropriate setup of a machine learning architecture and its pipeline was framed as a multi-attribute discussion-making problem. The proposed method was developed for a rapid occupant safety assessment with a particular supervised learning setup.
The proposed method consists of the decision-making preparation containing (i) the definition of an initial list of criteria and (ii) the review of them using sample alternatives, leading to (iii) the definition of the final criteria list. From the literature research, the PROMETHEE II decision-making method was selected. A version of the sorting-based algorithm proposed by Calders et al. was implemented.
The method was tested on data from a finite element model in the validation part. A final list of criteria was developed and used to rank sample alternatives resulting from a parameter variation. First tendencies of the influence of the alternative's parameters on its rank could be identified.
The method was discussed, and recommendations were derived. Overall, a high dependency on expert knowledge was identified. For the criteria, ordinal scales seemed apt. PROMETHEE II, with the sorting algorithm, delivered a plausible and distinct ranking, and the time complexity allowed the assessment of an immense number of alternatives simultaneously.
The method should be applied to a database based on a more realistic and validated finite element model. Further research will be dedicated to the vehicle characterisation for more than one dimension and to the dimensional reduction approach. The increasing need for efficient assessment methods will fuel further validation.

REFERENCES

[1] Ratingen, M.R. van, "Euro NCAP – from passive safety to assistance systems and beyond," *crash.tech 2022*, Ingolstadt, 2022.

[2] Östling, M. and Larsson, A., "Occupant Activities and Sitting Positions in Automated Vehicles in China and Sweden," *26th ESV*, Eindhoven, Netherlands, 2019.

[3] Reed, M.P. and Rupp, J.D., "An anthropometric comparison of current ATDs with the U.S. adult population," *Traffic Injury Prevention* 14(7):703–705, 2013, doi:10.1080/15389588.2012.752819.

[4] Wang, S.C., Hsieh, C.-H., Cheng, C.-T., Chiu, C.-H. et al., "Morphometric Characterisation of an Asian Reference Analytic Morphomics Population (A-RAMP)," *IRCOBI Conference*, Florence, Italy, 2019.

[5] Plaschkies, F., Vaculin, O., and Schumacher, A., "Assessment of the Influence of Human Body Diversity on Passive Safety Systems: A State-of-the-art Overview," *FISITA Web Congress*, doi:10.46720/F2021-PIF-071, 2021.

[6] Plaschkies, F., Vaculin, O., Pelisson, A., and Schumacher, A., "Schnelle Abschätzung des Crashverhaltens von Insassen unter Berücksichtigung der Vielfalt des Menschen: Robustheit, Datenintensität und Vorhersagekraft von Metamodellen," *VDI Fahrzeugsicherheit*, Berlin, doi:10.51202/9783181023877-313, 2022.

[7] Hwang, C.-L. and Yoon, K., "Multiple Attribute Decision Making: Methods and Applications A State-of-the-Art Survey," Springer eBook Collection, vol. 186, Springer, Berlin, Heidelberg, ISBN 978-3-642-48318-9, 1981.

[8] Zavadskas, E.K., Antucheviciene, J., and Kar, S., "Multi-Objective and Multi-Attribute Optimization for Sustainable Development Decision Aiding," *Sustainability* 11(11):3069, 2019, doi:10.3390/su11113069.

[9] Kahraman, C., "Fuzzy Multi-Criteria Decision Making," vol. 16, Springer US, Boston, MA, ISBN 978-0-387-76812-0, 2008.

[10] Wolny, M., "Analysis of the Multiple Attribute Decision Making Problem with Incomplete Information about Preferences among the Criteria," *MCDM* 11:187–197, 2016, doi:10.22367/mcdm.2016.11.12.

[11] Majdi, I., "Comparative evaluation of PROMETHEE and ELECTRE with application to sustainability assessment," Master Thesis, Concordia University, Montreal, Quebec, Canada, 2013.

[12] Linkov, I., Varghese, A., Jamil, S., Seager, T.P. et al., "Multi-Criteria Decision Analysis: A Framework for Structuring Remedial Decisions at Contaminated Sites," in: Linkov, I. and Ramadan, A.B. (eds.), *Comparative Risk Assessment and Environmental Decision Making*, Nato Science Series: IV: Earth and Environmental Sciences, Kluwer Academic Publishers, Dordrecht, ISBN 1-4020-1895-9:15–54, 2005.

[13] Winterfeldt, D. von and Edwards, W., "Decision analysis and behavioral research," Univ. Pr, Cambridge, ISBN 978-0521273046, 1986.

[14] Ayağ, Z., "An approach to evaluate CAM software alternatives," *International Journal of Computer Integrated Manufacturing* 33(5):504–514, 2020, doi:10.1080/0951192X.2020.1757156.

[15] Brans, J.P. and Vincke, P., "Note—A Preference Ranking Organisation Method," *Management Science* 31(6):647–656, 1985, doi:10.1287/mnsc.31.6.647.

[16] Almeida, A.T. de and Costa, A.P.C.S., "Modelo de decisão multicritério para priorização de sistemas de informação com base no método PROMETHEE," *Gest. Prod.* 9(2):201–214, 2002, doi:10.1590/S0104-530X2002000200007.

[17] Brans, J.P., Vincke, P., and Mareschal, B., "How to select and how to rank projects: The Promethee method," *European Journal of Operational Research* 24(2):228–238, 1986, doi:10.1016/0377-2217(86)90044-5.

[18] Akhavi, F. and Hayes, C., "A comparison of two multi-criteria decision-making techniques," *SMC'03 Conference Proceedings.*, Washington, DC, USA, IEEE, doi:10.1109/ICSMC.2003.1243938, ISBN 0-7803-7952-7:956–961, 2003.

[19] Keyser, W. de and Peeters, P., "A note on the use of PROMETHEE multicriteria methods," *European Journal of Operational Research* 89(3):457–461, 1996, doi:10.1016/0377-2217(94)00307-6.

[20] Calders, T. and Assche, D. van, "PROMETHEE is not quadratic: An O(qnlog(n)) algorithm," *Omega* 76:63–69, 2018, doi:10.1016/j.omega.2017.04.003.

[21] Pedregosa, F., Varoquaux, G., Gramfort, A., Verleysen, M. et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research* 12:2825–2830, 2011.

[22] The pandas development team, pandas-dev/pandas: Pandas, Zenodo, 2022.

[23] Marzougui, D., Samaha, R.R., Cui, C., Kan, C.-D. et al., "Extended Validation of the Finite Element Model for the 2010 Toyota Yaris Passenger Sedan," NCAC 2012-W-005, 2012.

[24] "Instrumentation for Impact Test - Part 1 - Electronic Instrumentation: SAE J211/1," in: *SURFACE VEHICLE RECOMMENDED PRACTICE*, 2014.

[25] Powers, D.M.W., "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," *Journal of Machine Learning Technologies* 2(1):37–63, 2011.

ACKNOWLEDGEMENT

The authors acknowledge the kind support of the German Academic Exchange Service (DAAD) in the project enGlobe, which supported the research in the frame of the Bavarian Center for Applied Research and technology with Latin America (AWARE).

APPENDIX

Table 4.
Initial list of criteria

Metamodel Setup Cost			
	Database Setup		
1	setup_db_num_sim_calibration_pprediction	Calibration simulations per environment (= per prediction)	
2	setup_db_num_sim_calibration_sum	Total number of calibration simulations in database	
3	setup_db_num_sim_crash	Total number of crash simulations in database	
4	setup_db_num_sim_occupant	Total number of occupant simulations in database	
5	setup_db_num_sim_assessment	Total number of samples used for metamodel assessment	
6	setup_db_num_sim_training	Total number of samples used for metamodel training	
7	setup_db_comp_time_calibration	Computation time for calibration simulations per prediction	
8	setup_db_comp_time_crash	Total computation time of all crash simulations in database	
9	setup_db_comp_time_occupant	Total computation time of all occupant simulations in database	
10	setup_db_comp_time_assessment	Total computation time of for assessment used samples	
	Training Phase		
11	setup_training_comp_time_metamodel	Computation time of metamodel's training	
12	setup_training_comp_time_assessment_sum	Computation time of metamodel's assessment on training data	
13	setup_training_comp_time_assessment_pprediction	Computation time for single prediction by metamodel	
14	setup_training_comp_time_calibration	Total computation time for calibration simulations used in training	
15	setup_training_calibration_sum	Number of calibration simulations used for training	
16	setup_training_calibration_pprediction	Calibration simulations per environment (= per prediction)	
17	setup_training_num_sim_crash	Total number of crash simulations used for training	
18	setup_training_num_sim_occupant	Total number of occupant simulations used for training	
19	setup_training_num_sim_assessment	Number of samples used for assessing in training phase (equals number of samples for training)	N/A
20	setup_training_comp_time_crash	Total computation time for crash simulations used for training	
21	setup_training_comp_time_occupant	Total computation time for occupant simulations used for training	
22	setup_training_comp_time_assessment	Total computation time of all for assessment used samples during training (equals computation time of simulations for training)	N/A
	Interpolation Assessment Phase		
23	setup_test_comp_time_assessment_sum	Computation time of metamodel's assessment (predictions)	
24	setup_test_comp_time_assessment_pprediction	Computation time for single prediction by metamodel	
25	setup_test_comp_time_calibration	Total computation time for calibration simulations used for assessment	
26	setup_test_calibration_sum	Number of calibration simulations used for assessment	
27	setup_test_calibration_pprediction	Calibration simulations per environment	
28	setup_test_num_sim_crash	Total number of crash simulations used for assessment	
29	setup_test_num_sim_occupant	Total number of occupant simulations used for assessment	
30	setup_test_num_sim_assessment	Total number of for assessment used samples	
31	setup_test_comp_time_crash	Total computation time for crash simulations used for assessment	
32	setup_test_comp_time_occupant	Total computation time for occupant simulations used for assessment	
33	setup_test_comp_time_assessment	Total computation time of all for assessment used samples (occupant & crash)	N/A
	Validity (Extrapolation) Assessment		
34	setup_val_comp_time_assessment_sum	Total computation time for predictions in extrapolation range	
35	setup_val_comp_time_assessment_pprediction	Computation time for single prediction by metamodel	
36	setup_val_comp_time_calibration	Total computation time for calibration simulations used for assessment	N/A
37	setup_val_calibration_sum	Number of calibration simulations used for assessment	N/A
38	setup_val_calibration_pprediction	Calibration simulations per environment	N/A
39	setup_val_num_sim_crash	Total number of crash simulations used for assessment	N/A
40	setup_val_num_sim_occupant	Total number of occupant simulations used for assessment	N/A
41	setup_val_num_sim_assessment	Total number of for assessment used samples	N/A
42	setup_val_comp_time_crash	Total computation time for crash simulations used for assessment	N/A
43	setup_val_comp_time_occupant	Total computation time for occupant simulations used for assessment	N/A
44	setup_val_comp_time_assessment	Total computation time of all for assessment used samples (occupant & crash)	N/A
Usage			
45	us_metamodel_num_sim_calibration	Calibration simulations per environment	
46	us_metamodel_time_sim_calibration	Computation time for calibration simulations per environment	

47	us_metamodel_time_prediction	Computation time for single prediction by metamodel	
48	us_prediction_type	Value of prediction type (binary classification to regression)	
49	us_prediction_outputs	Degree of detail of predictions (single value to full sensor time series)	
50	us_prediction_sensor_num	Number of not used sensors of available sensors in dummy	
51	us_prediction_sensor_relevance	Relevance of used sensors (irrelevant to utilized in legislation)	
52	us_prediction_anthropometrics	Detail of degree of anthropometrical distinction	
53	us_MLmetric	Value from assessed metric	
Validity Range			
54	val_range	Width of validity range	
55	val_range_retraining	Cost to retrain the metamodel	N/A

N/A – not implemented

Table 5
List of final criteria

	Name	**Description**	
1	setup_db_num_sim_crash	Total number of crash simulations in database	
2	setup_db_comp_time_crash	Total computation time of all crash simulations in database	
3	setup_db_num_sim_assessment	Total number of samples used for metamodel assessment (test & validation)	
4	setup_db_num_sim_training	Total number of samples used for metamodel training	
5	setup_training_calibration_sum	Number of calibration simulations used for training	
6	setup_training_comp_time_assessment_ppprediction	Computation time for single prediction by metamodel	
7	setup_training_comp_time_assessment_sum	Computation time of metamodel's assessment (predictions) on training data	
8	setup_training_comp_time_calibration	Total computation time for calibration simulations used in training	
9	setup_training_comp_time_crash	Total computation time for crash simulations used for training	
10	setup_training_comp_time_metamodel	Computation time of metamodel's training	
11	setup_training_num_sim_crash	Total number of crash simulations used for training	
12	setup_training_num_sim_occupant	Total number of occupant simulations used for training	
13	setup_test_comp_time_crash	Total computation time of all crash simulations in database	
14	setup_test_num_sim_crash	Total number of crash simulations used for assessment	
15	setup_test_comp_time_assessment_sum	Computation time of metamodel's assessment (predictions)	
16	setup_test_calibration_sum	Number of calibration simulations used for assessment	
17	setup_test_num_sim_occupant	Total number of occupant simulations used for assessment	
18	setup_test_comp_time_assessment_ppprediction	Computation time for single prediction by metamodel in test phase	
19	setup_val_comp_time_assessment_ppprediction	Computation time for single prediction by metamodel in validation phase	
20	setup_val_comp_time_assessment_sum	Total computation time for predictions in extrapolation range	
21	us_MImetric	Value from assessed metric; F-score for classification / R^2-score for regression	
22	us_prediction_anthropometrics	Detail of degree of anthropometrical prediction, grades, where 1 is best	
		5	1 percentile
		4	2 percentiles
		3	3-4 percentiles
		2	≥ 5 percentiles
		1	Anthropometrical parameter
23	us_prediction_outputs	Degree of detail of predictions, grades, where 1 is best	
		3	Single value
		2	Relevant characteristics
		1	Full sensor time series
24	us_prediction_sensor_num	Number of not used sensors (reference are available sensors of used dummy)	
25	us_prediction_sensor_relevance	Relevance of used sensors, grades, where 1 is best	
		4	Irrelevant
		3	Physics relevant
		2	Utilized in consumer tests
		1	Utilized in legislation
26	us_prediction_type	Value of prediction type, grades, where 1 is best	
		5	Binary classification (e. g. critical, uncritical)
		4	3 classes
		3	4-5 classes
		2	≥ 6 classes
		1	Regression
27	us_metamodel_num_sim_calibration	Calibration simulations per environment	
28	val_range	Width of validity range	