INTERNATIONAL DESIGN CONFERENCE – DESIGN 2024
https://doi.org/10.1017/pds.2024.228

A conceptual MCDA-based framework for machine learning algorithm selection in the early phase of product development

Sebastian Sonntag ✉, Erik Pohl, Janosch Luttmer, Jutta Geldermann and Arun Nagarajah

University of Duisburg-Essen, Germany

✉ sebastian.sonntag@uni-due.de

Abstract

Despite the potential to enhance efficiency and improve quality, AI methods are not widely adopted in the context of product development due to the need for specialized applications. The necessary identification of a suitable machine learning (ML) algorithm requires expert knowledge, often lacking in companies. Therefore, a concept based on a multi-criteria decision analysis is applied, enabling the identification of a suitable ML algorithm for tasks in the early phase of product development. The application and resulting advantages of the concept are presented through a practical example.

Keywords: *artificial intelligence (AI), machine learning, product development, decision making, multi-criteria decision aiding (MCDA)*

1. Introduction

Artificial intelligence (AI) has made great progress at a tremendous speed in recent years. For instance, the number of patent applications tripled compared to 2015 (Zhang et al., 2022). Due to their ability to reduce costs and time while continuously improving quality, AI methods are increasingly attracting the attention of companies (Diemer et al., 2020).

Those advantages become especially evident in the context of product development, where the trend towards individual products, rising cost pressures and shorter innovation cycles presents the challenge of enhancing efficiency, reducing time to market, and improving quality to maintain competitiveness in the market (Krause, 2018). Due to its significant impact on overall development time and costs, the early phase of product development is particularly relevant in this context (Ehrlenspiel et al., 2014). Hence, a range of existing approaches, from defining requirements (Christensen et al., 2017) to assisting in the design process (Camburn et al., 2020), substantiate the supportive role of applying machine learning (ML) as subfield of AI in this context. However, recent studies show limited adoption of these ML algorithms within companies (Reder, 2021).

In terms of causative factors, companies struggle to identify appropriate activities for the application of ML support. Moreover, it becomes evident that solutions customized to their particular requirements are needed. This raises the issue of identifying appropriate ML algorithms, a task that demands expert knowledge not readily available within the companies. (Reder, 2021) In addition, the process of identifying suitable ML algorithms for a particular problem is characterised by a lengthy trial-and-error process that involves testing different ML algorithms. As the current process relies on expert knowledge, it is highly subjective. (Sarang, 2023)

Therefore, the question arises:

How can individuals who lack experience be aided in identifying an appropriate ML algorithm for their particular task in the context of product development, taking into account their specific preferences and constraints?

1.1. Elaboration of the requirements

To support ML algorithm selection, an approach should meet various requirements derived from current challenges of implementing ML in company processes.

Companies are currently unaware of potential use cases for ML and the specific problems that can be addressed with particular ML algorithms (Reder, 2021). Consequently, the approach shall be capable to deduce potential ML algorithms suitable for addressing a specific task (R1). Building on this, early decision making shall be supported, even in cases where datasets for the task may not be available yet or are to be generated with regards to a specific ML algorithm (R2).

Companies seeking to implement ML in their processes aim to address individual tasks tailored to their specific constraints and preferences (Reder, 2021). Hence, the approach shall accommodate the individual preferences of the companies in terms of the evaluation criteria (R3) and address the constraints that must be taken into consideration when using ML algorithms (R4). The latter becomes especially important when the performance measures of multiple ML algorithms are nearly equal and cannot serve as decision criteria.

Many companies, especially small and medium-sized ones, lack expertise in the field of ML (Reder, 2021). Consequently, the approach shall be applicable without the need for prior ML knowledge (R5). To support the decision-making process, the approach shall be transparent in terms of the evaluation process and provide results that can be analysed and comprehended, allowing for the possibility of questioning, or adjusting one's preferences (R6).

1.2. Related work

Existing approaches that aim to support identifying a suitable ML algorithm can be classified into four distinct categories.

Quantitative criteria (1)

Numerous publications compare a variety of ML algorithms using performance measures like precision, recall or the F1 score (Luttmer et al., 2023). These assessments are often conducted for specific ML algorithms, tailored to specific tasks or datasets. For instance, Abo et al. (2021) evaluate various supervised learning algorithms in the use case of Arabic sentiment analysis. A more general approach by Blagec et al. (2022) focuses on building a knowledge graph containing benchmark results and performance analysis for a range of ML algorithms.

AutoML (2)

The term AutoML summarizes a variety of approaches, in which the selection of an appropriate ML algorithm is treated as an optimization problem. The hyperparameters of supported ML algorithms are iteratively adjusted based on an existing dataset of the specific problem. Performance measures serve as evaluation criteria to identify the most suitable algorithm. Common examples are Auto-Sklearn and Auto-Weka. (Waring et al., 2020)

Qualitative criteria (3)

Several approaches define lists of criteria comprising qualitative measures for comparing multiple ML algorithms. Commonly those criteria lists are limited to supervised or unsupervised learning and either created or provided within the context of specific application areas. For instance, Lickert et al. (2021) provide such selection criteria for a limited set of supervised learning algorithms in the context of reverse logistic, while Riesener et al. (2020) offer criteria for a limited number of ML algorithms in the context of product development. The evaluation of the ML algorithms includes criteria such as transparency, tolerance for erroneous and irrelevant values, and computational efficiency.

Knowledge representations (4)

Further approaches focus on the creation of knowledge databases, that facilitate the finding of a suitable ML algorithm, along with associated insights into its advantages, disadvantages, and potential applications, based on a problem formulation. Gerschütz et al. (2021) introduce a semantic web-based approach, which includes the characterization of ML algorithms in the context of potential use cases in product development. Gerschütz et al. (2023) present an ontology-based approach, that enables the identification of an appropriate ML algorithm using query syntax.

2. Research goal

The comparison of the elaborated requirements and existing approaches yields the results depicted in Table 1.

Table 1. Derived requirements for the approach and their fulfilment by existing approaches

ID	Requirement	(1)	(2)	(3)	(4)
R1	The approach shall be able to deduce a suitable ML algorithm for a predefined task.	○	○	○	●
R2	The approach shall be applicable without preexisting datasets.	●	○	●	●
R3	The approach shall provide the opportunity to express individual preferences based on the importance of criteria.	●	●	●	○
R4	The approach shall consider, regarding the evaluation criteria, the constraints that limit the application of ML algorithms.	○	○	●	○
R5	The approach shall be applicable without prior knowledge regarding ML.	○	●	○	●
R6	The approach shall be transparent regarding the evaluation process and results.	○	○	○	●

●: fulfilled ○: not fulfilled

R1 and R2: While existing quantitative and qualitative lists of criteria (1, 3) offer evaluation measures for supervised and unsupervised ML algorithms, they do not address the extent to which a given ML algorithm can even address the underlying task. For instance, a task deduced to a classification problem cannot be addressed by a ML algorithm designed for clustering. AutoML-related approaches (2) also face the challenge of requiring a preexisting dataset for evaluating a suitable ML algorithm.
R3 and R4: Quantitative criteria lists and AutoML-related approaches (1, 2) are limited to performance metrics. Although qualitative criteria lists (3) address constraints, there is a notable lack of consistency in these criteria, particularly in terms of incompleteness and their relevance to specific application domains. For instance, Lickert et al. (2021) define the criterion *number of features*, which is not covered in the elaborated criteria list for all ML algorithms. In contrast, knowledge-based approaches (4) do not explicitly state boundary conditions directly associated with the respective ML algorithms, nor do they facilitate the definition of individual preferences regarding the selection process.
R5 and R6: Quantitative as well as qualitative criteria lists (1, 3) do not give guidance on how to apply the criteria to a given problem statement in order to select an appropriate ML algorithm. For instance, the criterion presented by Kotsiantis et al. (2006) regarding *tolerance to highly interdependent attributes* can be challenging for non-experts to understand, making it difficult to assess its significance. In addition, no evaluation method is presented regarding the evaluation process. Thereby, with respect to AutoML-related approaches (2), the evaluation process can be described as a black box (Sarang, 2023).
Table 1 indicates that none of the existing approaches can fully meet the elaborated requirements and cannot be effectively adapted to support the selection of ML algorithms in the context of product development. Therefore, they are unable to provide seamless decision-making support, spanning from task definition to evaluation and the provision of results in an understandable format. For such highly complex decision-making processes, where a multitude of individual conditions and preferences need to be considered, evaluated and the results analysed, multi-criteria decision aiding (MCDA) methods have proven to be a robust solution (Brans and Smet, 2016).
Therefore, the goal of this paper is to develop a MCDA-based concept that meets the requirements, thus aiding in the identification of suitable ML algorithms in the field of product development. This approach should consider one's specific constraints and preferences, all without requiring expert knowledge in ML.

3. Development of the concept

Based on the requirements outlined in Table 1, a concept can be developed, to facilitate the decision-making process for selecting suitable ML algorithms for applications in the context of product

https://doi.org/10.1017/pds.2024.228 Published online by Cambridge University Press

development in a systematic and objective manner. The concept can be divided into three distinct phases, each with the goal of narrowing down the solution space, as depicted in Figure 1.

Phase 1: Derivation of the required ML capability.

The starting point can be a problem formulation or an activity within the context of product development for which an appropriate ML algorithm is sought for application (Sonntag et al., 2023). At first, the objective is to identify suitable ML algorithms capable of handling the core activity underlying the domain-specific problem, such as whether it involves classification or clustering. This helps to narrow the solution space, for instance from general ML algorithms, to either supervised or unsupervised ones. Therefore, a reference process is established (R1, R5).

Phase 2: Evaluating the ML algorithms.

Once the solution space has been reduced to a specific class of ML algorithms, the remaining ones must be evaluated based on the user's preferences and individual constraints. This can be modelled as a MCDA problem, and therefore solved using an appropriate MCDA method (R2, R3, R4).

Phase 3: Analysing the results and making the final decision.

The evaluation of the ML algorithms may yield multiple candidates that align the company's preferences, among which one appears to be superior to the rest. A crucial step relies on the analysis of the result to understand the reason for the superiority, leading to the decision of the ML algorithm best aligns with one's preferences. To effectively support this process, several diagrams providing insights in the decision-making process are deduced to make the process as comprehensive as possible (R6).

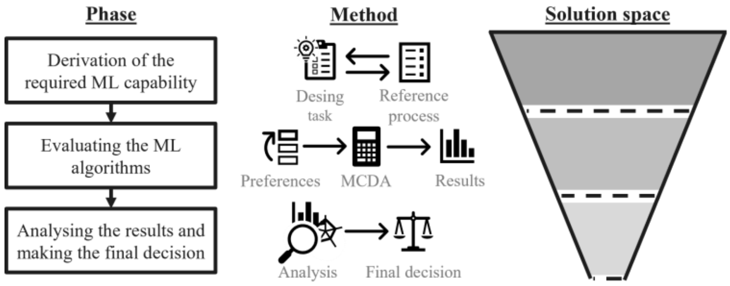


Figure 1. Concept for the decision support for the ML algorithm selection process

To apply the methods corresponding to the phases, the underlying prerequisites must be developed. Therefore, in section 3.1, the development of the reference process will be addressed, and in section 3.2, the evaluation matrix will be established as a building block for implementing the MCDA method, which will be further discussed in section 3.3. This sets the stage for applying the concept to an example use case, discussed in section 4.

3.1. Definition of the reference process

ML algorithms can be categorized into three distinct classes: supervised, unsupervised and reinforcement learning. As indicated by Riesener et al. (2020), most tasks in product development can be addressed through supervised and unsupervised learning, making them the focus of the concept.

Supervised learning algorithms establish the relationship between provided input data and their corresponding outputs, enabling them to perform prediction tasks (Jung, 2022). These can be distinguished between classification (Cl), which deals with the prediction of a category, and regression (R), which deals with the prediction of a numerical value (Bramer, 2020).

Unsupervised learning algorithms handle unlabelled data, where the desired output is not known beforehand. Consequently, the objective is to identify patterns or similarities within the provided input data. (Jung, 2022) These algorithms can be further classified based on their ability to derive association rules (Ar), which involves identifying relationships between given variables in the form of rules, and clustering (Clu), which involves identifying similar data or grouping them into clusters (Bramer, 2020).

To enable the identification of the class of algorithms capable of addressing a domain specific task within the context of product development, a reference process is established.

The definition of the reference process is based on VDI 2221, which provides a comprehensive framework for design phases with the aim to be applicable to a wide range of companies (Bender and

https://doi.org/10.1017/pds.2024.228 Published online by Cambridge University Press

Gericke, 2021). Due to its crucial role in the whole development process (Ehrlenspiel et al., 2014), the reference process focuses on the early phase of product development, resulting in the definition of four phases. The reference literature mentioned in VDI 2221 was consulted to further elaborate the reference process with sub-steps and corresponding activities. The excerpt of the resulting reference matrix in Table 2 illustrates an exemplary exclusive assignment based on the characteristics of the tasks and potentially suitable ML algorithms. For instance, by categorizing requirements into firm requirements and desires, the class is predefined, favouring a supervised classification algorithm.

Table 2. Reference matrix to map the reference process

Phase	Sub step	Activity	Cl	R	Ar	Clu
Clarification of problem or task	Elaborate requirements	Information extraction from engineering documents	●			
	Specify requirements	Identification of attributes connected to the requirement	●			
	Structuring requirements	Categorisation of the requirements	●			
	Analyse requirements	Identification of conflicting objectives				●
Determination of functions and their structure	Determine functions	Identification of functions derived from the requirements	●			
	Structuring functions	Categorisation of the functions	●			
	Function analysis	Identification of interrelationships				●
Search for solution principles and their structures	Identify solutions	Identification of vital characteristics and principles in existing solutions				●
	Structure solution space	Categorisation of principal solutions	●			
	Form overall solution concepts	Identification of compatibilities between principal solutions			●	
	Limit the solution space	Group similar solutions				●
Assessment and selection of solution concepts	Define assessment criteria	Extract assessment criteria based on requirements	●			
	Analyse assessment criteria	Identify contradictions and mergeable criteria				●
	Define evaluation metrics	Define measures for properties		●		
	Specify assessment values	Evaluate concepts		●		

Cl: Classification, R: Regression, Ar: Association rule, Clu: Clustering

3.2. Elaboration of evaluation criteria

The evaluation matrix, serving as the foundation for applying a MCDA method, has been constructed based on the generic steps for conducting an evaluation outlined by Bender and Gericke (2021). Hence, the criteria (C) are defined based on the constraints arising from ML algorithm application in the early phase of product development and within the company itself. The criteria aim to be mutually exclusive and encompass distinguishable characteristics of the ML algorithms. Therefore, aspects such as implementation complexity, which is directly related to factors like parameter complexity, can be evaluated alongside them. Due to their strong data relationship and individual adjustments, performance measures like accuracy or precision are not considered.

The analysed ML algorithms for the concept are based on Schmid et al. (2021), who provide a structured overview of commonly used ML algorithms. Additionally, K nearest neighbours and three ML algorithms covering Ar are considered, based on their relevant application (Riesener et al., 2020). This results in a total of eight supervised and eight unsupervised ML algorithms to be included in the concept. The essential information required to define the features for criteria evaluation were collected through a systematic literature review. In total, 164 publications, including theoretical books e.g. (Jung, 2022), reviews and surveys e.g. (Kotsiantis et al., 2006), as well as applications of the presented ML algorithms e.g. (Licen et al., 2023) were examined. This leads to the evaluation matrix presented in Table 3.

https://doi.org/10.1017/pds.2024.228 Published online by Cambridge University Press

C1: Data flexibility (max. 3)
Depending on the design task, different types of data may be available, such as design parameters (continuous) or requirements (discrete). Matching data types with the ML algorithm is important to avoid data transformations. One can distinguish between discrete (1), continuous (2) or both (3). Transforming discrete values in continuous ones is a more difficult process, making it less favourable.
C2: Data quantity (max. 4)
The amount of available data can vary, especially in the early phase, depending on the design task. For instance, requirements elaboration may have abundant data compared to creative tasks. The quantity of data significantly influences the ML algorithm applicability. Therefore, the criterion evaluates supervised learning algorithms based on the required training data, favouring less. For unsupervised learning algorithms, the flexibility in data amount is assessed, with arbitrary sizes preferred.
C3: Data preprocessing (max. 4)
The quality of data in design-related tasks can vary. Different ML algorithms vary in the effort required to prepare the data for use. Several characteristics should be considered, such as the requirement of nominal values, the ability to handle noise and high-dimensional data. In this context, the number of combinations of these supported by the ML algorithm is assessed.

Table 3. Evaluation matrix derived for ML algorithms (values range from 1 to 3 or 4; higher values indicate a better performance)

Task	Algorithm	C1	C2	C3	C4	C5	C6	C7	C8	C9
Cl/R	K nearest neighbours (KNN)	2	3	1	3	3	2	3	3	2
Cl/R	Multilayer perceptron (MLP)	2	1	2	1	3	3	1	2	1
Cl/R	Convolutional neural network (CNN)	2	1	2	1	3	4	2	1	1
Cl/R	Long short-term memory (LSTM)	2	1	3	2	3	2	1	2	1
Cl/R	Decision tree (DT)	3	4	3	4	2	3	2	2	4
Cl/R	Random forest (RF)	3	4	4	3	2	4	3	3	3
Cl/R	Support vector machine (SVM)	2	4	2	2	1	3	2	4	2
Cl/R	Naïve Bayes (NB)	1	4	3	4	3	2	4	2	3
Clu	K-means (Km)	2	4	2	2	2	3	4	1	3
Clu	Hierarchical clustering (HC)	3	3	1	4	1	1	2	2	4
Clu	DBSCAN	2	4	2	3	2	3	2	3	3
Clu	Self-organizing map (SOM)	2	4	3	1	1	4	2	4	4
Clu	Fuzzy c-means (Fcm)	2	3	3	2	2	2	3	1	3
Ar	Apriori	1	3	2	3	1	2	2	3	3
Ar	ECLAT	1	3	2	3	1	3	3	3	3
Ar	FP growth (FPG)	1	3	2	2	1	3	3	2	4

C4: Complexity of parameter definition (max. 4)
The implementation of a ML algorithm can become a challenging task depending on the number and complexity of hyperparameters that need to be adjusted. The quantity or complexity of parameters or functions to be defined can range from minimal (1) to few (2), moderate (3), or extensive (4).
C5: Scalability (max. 4)
An important aspect to consider when evaluating ML algorithms is the computational effort, which determines the hardware requirements needed to compute the resulting models. An objective assessment of the computational complexity of an algorithm can be achieved using the O-notation. Constant complexity is preferred (4), while exponential complexity is to be avoided (1).
C6: Memory requirements (max. 4)
The required memory for a ML algorithm can lead to extra hardware expanses. The memory requirements can be quantified similarly to scalability, using space complexity in O-notation.
C7: Time costs (max. 4)
To improve efficiency through the implementation of a ML algorithm, time is a critical factor. The criterion is assessed based on the time required for model training and execution.

https://doi.org/10.1017/pds.2024.228 Published online by Cambridge University Press

C8: Robustness (max. 4)
In critical design activities, such as the requirement definition, it is crucial for the ML algorithm to provide reliable and consistently excellent support. Characteristics contributing to this include tolerance to changes in input data and the absence of significant error propagation. The value is determined based on the number of characteristics covered by the ML algorithm with a maximum of four.
C9: Understandability (max. 4)
For certain design tasks, understanding the decision-making process of a ML algorithm can be crucial. A distinction is made between black box models (1), models with difficult-to-comprehend (2) and easy-to-comprehend formulas (3) and models with a visual representation of the decision process (4).

3.3. PROMETHEE for selecting ML algorithms

MCDA is a methodical approach used to evaluate multiple conflicting criteria in decision-making contexts (Belton and Stewart, 2002). In the present decision problem, MCDA can be leveraged to determine the most suitable ML algorithm from a finite set of potential candidates. The product design literature has traditionally emphasized utility analysis for this purpose. Nevertheless, utility analysis has well-known limitations, despite broad usage (Greco et al., 2019). This method assumes linear, compensatory behaviour among criteria, which may not accurately represent the complexities of real-world decision-making. Furthermore, it can be influenced by the specific utility functions chosen, resulting in possible inconsistencies in outcomes (Greco et al., 2019).
Due to these limitations, outranking methods like PROMETHEE have been developed. PROMETHEE offers various benefits over traditional utility analysis, primarily the capability to establish a partial preorder of alternatives. This permits incomparability between alternatives, allowing for a more detailed and comprehensive analysis of the decision problem (Brans and Smet, 2016). This capability is particularly useful when evaluating the relative positioning of alternatives, rather than simply determining the best option. PROMETHEE allows decision-makers to perform sensitivity analyses that determine the impact of changes in input parameters or weights on their decisions (Mareschal, 1988).
Besides the evaluation table representing the alternative's performance under each criterion (refer to Table 3), PROMETHEE II requires a weighting and preference function for every criterion. The PROMETHEE net flow is then given by

$$\phi(a) = \phi^+(a) - \phi^-(a) = \frac{1}{n-1}\sum_{j=1}^{k} \sum_{x\in A} \left[P_j(a,x) - P_j(x,a)\right]w_j = \frac{1}{n-1}\sum_{j=1}^{k} \phi_j(a)w_j, \quad (1)$$

where $P_j(a,b)$ is the preference function for criterion j applied to alternatives a and b. There are six preference functions commonly used in PROMETHEE, selected based on the decision maker's preferences (Brans and Smet, 2016).
For outranking methods such as PROMETHEE, the criteria weights reflect criterion importance. This can be determined more easily by decision makers using objective evaluations. The SIMOS method (Figueira and Roy, 2002) is a commonly employed approach to determine criterion importance, where the decision maker arranges a set of cards representing the criteria based on their preferences.

4. Application of the concept

The developed concept will be demonstrated using the example task of information extraction from engineering documents (Luttmer et al., 2023). As selecting an appropriate ML algorithm based solely on performance measures may not be adequate, this case provides an opportunity to demonstrate the benefits of the concept.

4.1. Identification of suitable ML algorithms and evaluation (Phase 1 and 2)

In Phase 1, the class of ML algorithms capable of solving the task is determined by identifying the activity in the reference matrix provided in Table 2. This allows the delimitation of the solution space to ML algorithms capable of performing classification, which need to be further evaluated in Phase 2. Here, the initial step is to define the weights that represent the user's preferences for evaluation criteria. Hence, the SIMOS method (Figueira and Roy, 2002), introduced in Chapter 3.3, is utilized, assuming potential preferences based on the nature of the underlying task. For instance, it is assumed that a vast

https://doi.org/10.1017/pds.2024.228 Published online by Cambridge University Press

amount of engineering documents respectively training data, is available, which implies that the preprocessing effort should be minimal. Since the identification of the requirements is a critical step, robustness is considered the most important criterion. The resulting weights, as well as the defined input parameters to assess the ML algorithms using the PROMETHEE method, are outlined in Table 4. The preference information, subject to potential adjustments, was selected by a decision maker to demonstrate the methodology. We utilized PROMETHEE-Cloud[1] software to evaluate the current decision model, as it presents numerous sensitivity analyses. (Pohl and Geldermann, n.d.).

Table 4. Input parameters for the evaluation with PROMETHEE

Parameters	C1	C2	C3	C4	C5	C6	C7	C8	C9
Criteria weights	0,146	0,041	0,146	0,116	0,131	0,116	0,086	0,162	0,056
Orientation	max	max	max	max	max	max	max	max	max
Preference function	usual	usual	level	usual	usual	level	gaussian	usual	level
Preference function parameter			q = 1			q = 1	$\sigma = 3$		q = 1

4.2. Analysis of evaluation results and final decision (Phase 3)

The results indicate that the decision maker prefers RF followed by DT and NB. Figure 2 a) shows the resulting PROMETHEE flows for every alternative. The $\Phi^+(a)$ flow is a measure of how much other alternatives are dominated by alternative a and the $\Phi^-(a)$ flow is a measure of how much alternative a is dominated by others. The higher the resulting net flow _PHI_ (a), the better the alternative according to the preference structure of the decision maker. The partial preorder by PROMETHEE allow for a more nuanced analysis of the decision problem. Figure 2 b) shows the partial preorder graph, where two nodes are connected, if the first alternative is preferred over the second. The partial preorder graph yields the same results for the most preferred alternative RF, followed by DT and NB. Moreover, NB is preferred over SVM, LSTM and KNN which are incomparable according to the preferences of the decision maker. Ultimately, the least preferred alternatives MLP and CNN are also incomparable.

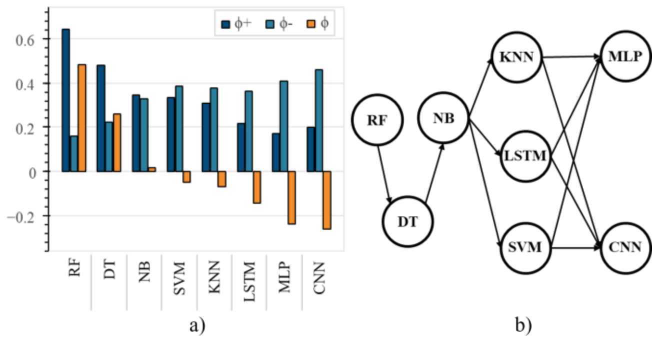


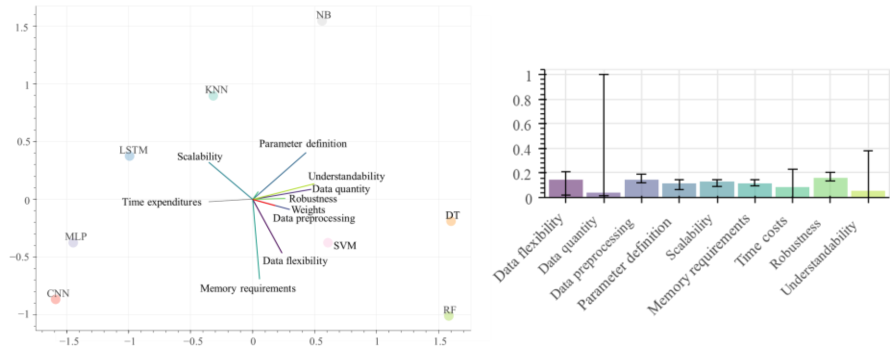
Figure 2. a) PROMETHEE flows for the alternatives; b) partial preorder graph

The PROMETHEE GAIA plane (see Figure 3a) provides a visual presentation of the decision problem, in which criteria and weights are represented through vectors, and alternatives are represented as points on the plane. This plane is generated through principal component analysis of the uni criterion net flow matrix $(\Phi_j(a))_{j,a}$ defined in Eq. 1 (Brans and Smet, 2016). When criteria vectors point in the same direction, they support alternatives located in that direction. The decision maker's preference, represented by the weight vector, also favours alternatives in that direction. Based on Figure 3 a), it is evident that RF and DT are supported by most of the criteria as well as the weight vector. Here, the length of the vector indicates the support of an alternative by a criterion (Brans and Smet, 2016).
Weight stability intervals in MCDA are ranges where the criteria weights can vary without affecting the overall ranking of the alternatives. This provides insight into the decision's robustness when changes occur in the weights assigned to different criteria (Mareschal, 1988). The decision problem's weight stability intervals are displayed in Figure 3 b), where the current weighting is shown as a bar graph

[1] https://promethee.pom.uni-due.de

https://doi.org/10.1017/pds.2024.228 Published online by Cambridge University Press

together with the intervals. The criterion data quantity has therefore no influence on the current ranking since a weighting between 0 and 1 will not change the ranking of the alternatives. The ranking is robust in terms of data flexibility and understandability. However, it is less stable for time costs compared to data flexibility and understandability. However, the ranking is not highly robust regarding the weights of the other criteria and a small alteration in the criteria weight can lead to different results.



a) b)

Figure 3. a) PROMETHEE GAIA plane; b) weight stability intervals

5. Conclusion and outlook

Identifying a suitable ML algorithm to address a given task is a crucial step in adopting ML algorithms for product development. Conventional procedures typically necessitate pre-existing data sets and depend on subjective expert knowledge, which may not be accessible for many companies.

Therefore, a three-phase decision support concept has been developed. First, the design-related task is translated into a ML-related one to identify a class of ML algorithms capable of solving the task. Second, the multi-criteria decision support approach PROMETHEE is employed for the analysis, which criteria address the essential constraints for ML algorithm implementation in design-related tasks and companies, while also considering individual preferences. Third, the result of the evaluation is inspected via graphical representation and sensitivity analysis to comprehend the decision-making process. The examined use case shows, that commonly used ML algorithms for the discussed task got recommended, thus analysing the decision problem using the PROMETHEE methods yields valid results. In addition, the approach allows a consideration and clear distinction of the preferred ML algorithms regarding qualitative criteria, resulting in a systematic, comprehensive, and objective decision support.

Future work will focus on enhancing and extending the concept. Therefore, additional ML algorithms, especially those related to reinforcement learning, will be analysed, and evaluated to expand the evaluation matrix. Further expansion will encompass the established reference process to cover the entire development process. In addition, the applicability of the criteria by non-experts will be investigated and improved through conducting case studies.

References

Abo, M.E.M., Idris, N., Mahmud, R., Qazi, A. (2021), “A Multi-Criteria Approach for Arabic Dialect Sentiment Analysis for Online Reviews: Exploiting Optimal Machine Learning Algorithm Selection”, *Sustainability*, Vol. 13 No. 18, p. 10018. https://doi.org/10.3390/su131810018

Belton, V. and Stewart, T.J. (2002), *Multiple Criteria Decision Analysis: An Integrated Approach*, Springer US, Boston, MA, s.l. https://doi.org/10.1007/978-1-4615-1495-4_11

Bender, B. and Gericke, K. (Eds.) (2021), *Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung*, 9th ed. 2021, Springer, Berlin. https://doi.org/10.1007/978-3-662-57303-7

Blagec, K., Ott, S. (2022), “A curated, ontology-based, large-scale knowledge graph of artificial intelligence tasks and benchmarks”, *Scientific Data*, Vol. 9 No. 1, p. 322. https://doi.org/10.1038/s41597-022-01435-x

Bramer, M. (2020), *Principles of Data Mining*, *Undergraduate Topics in Computer Science*, 4th ed. 2020, Springer London. https://doi.org/10.1007/978-1-4471-7493-6

Brans, J.-P. and Smet, Y. de (2016), “PROMETHEE Methods”, in Greco, S., Ehrgott, M. (Eds.), *Multiple criteria decision analysis: State of the art surveys, International Series in Operations Research & Management Science*, Vol. 233, Springer, New York, pp. 187–219. https://doi.org/10.1007/978-1-4939-3094-4

https://doi.org/10.1017/pds.2024.228 Published online by Cambridge University Press

Camburn, B., Arlitt, R., Anderson, D., Sanaei, R., Raviselam, S., Jensen, D. and Wood, K.L. (2020), “Computer-aided mind map generation via crowdsourcing and machine learning”, *Research in Engineering Design*, Vol. 31 No. 4, pp. 383–409. https://doi.org/10.1007/s00163-020-00341-w

Christensen, K., Nørskov, S., Frederiksen, L. and Scholderer, J. (2017), “In Search of New Product Ideas: Identifying Ideas in Online Communities by Machine Learning and Text Mining”, *Creativity and Innovation Management*, Vol. 26 No. 1, pp. 17–30. https://doi.org/10.1111/caim.12202

Diemer, J., Elmer, S., Gaertler, M. and Gamer, T. (2020), KI in der Industrie 4.0: Orientierung, Anwendungsbeispiele, Handlungsempfehlungen, Bundesministerium für Wirtschaft und Energie (BMWi)

Ehrlenspiel, K., Kiewert, A., Lindemann, U. and Mörtl, M. (2014), *Kostengünstig Entwickeln und Konstruieren: Kostenmanagement bei der integrierten Produktentwicklung*, VDI-Buch, 7. Aufl. 2014, Springer Berlin Heidelberg, Berlin. https://doi.org/10.1007/978-3-642-41959-1

Figueira, J.R. and Roy, B. (2002), “Determining the weights of criteria in the ELECTRE type methods with a revised Simos' procedure”, *European Journal of Operational Research*, Vol. 139 No. 2, pp. 317–326. https://doi.org/10.1016/S0377-2217(01)00370-8

Gerschütz, B., Goetz, S. and Wartzack, S. (2023), “AI4PD—Towards a Standardized Interconnection of Artificial Intelligence Methods with Product Development Processes”, *Applied Sciences*, Vol. 13 No. 5, p. 3002. https://doi.org/10.3390/app13053002

Gerschütz, B., Schleich, B. and Wartzack, S. (2021), “A semantic web approach for structuring data-driven methods in the product development process”, *DS 111: Proceedings of the 32nd Symposium Design for X (DFX2021)*, 2021, pp. 1-10. https://doi.org/10.35199/dfx2021.15

Greco, S., Ishizaka, A., Tasiou, M. and Torrisi, G. (2019), “On the Methodological Framework of Composite Indices: A Review of the Issues of Weighting, Aggregation, and Robustness”, *Social Indicators Research*, Vol. 141 No. 1, pp. 61–94. https://doi.org/10.1007/s11205-017-1832-9

Jung, A. (2022), *Machine Learning: The Basics, Machine Learning*, 1st ed. 2022, Springer Singapore, Singapore. https://doi.org/10.1007/978-981-16-8193-6

Kotsiantis, S.B., Zaharakis, I.D. and Pintelas, P.E. (2006), “Machine learning: a review of classification and combining techniques”, *Artificial Intelligence Review*, Vol. 26 No. 3, pp. 159–190. https://doi.org/10.1007/s10462-007-9052-3

Krause, D. (2018), *Methodische Entwicklung modularer Produktfamilien: Hohe Produktvielfalt beherrschbar entwickeln*, Springer Berlin Heidelberg, Berlin. https://doi.org/10.1007/978-3-662-53040-5

Licen, S., Astel, A. and Tsakovski, S. (2023), “Self-organizing map algorithm for assessing spatial and temporal patterns of pollutants in environmental compartments: A review”, *The Science of the total environment*, Vol. 878, p. 163084. https://doi.org/10.1016/j.scitotenv.2023.163084

Lickert, H., Bilge, P. (2021), “Selection of Suitable Machine Learning Algorithms for Classification Tasks in Reverse Logistics”, *Procedia CIRP*, Vol. 96, pp. 272–277. https://doi.org/10.1016/j.procir.2021.01.086

Luttmer, J., Prihodko, V., Ehring, D. and Nagarajah, A. (2023), “Requirements extraction from engineering standards – systematic evaluation of extraction techniques”, *Procedia CIRP*, Vol. 119, pp. 794–799. https://doi.org/10.1016/j.procir.2023.03.125

Mareschal, B. (1988), “Weight stability intervals in multicriteria decision aid”, *European Journal of Operational Research*, Vol. 33 No. 1, pp. 54–64. https://doi.org/10.1016/0377-2217(88)90254-8

Pohl, E. and Geldermann, J. (n.d.), “Supporting environmental decisions using PROMETHEE-Cloud: A web app to support multi-criteria decisions”, *Submitted to Journal of Environmental Modelling & Software*.

Reder, B. (2021), *IDG Studie Machine Learning* 2021. [online] Available at: https://www.lufthansa-industry-solutions.com/de-de/studien/idg-studie-machine-learning-2021 (accessed 14.11.2023).

Riesener, M., Doelle, C. (2020), “Identification of evaluation criteria for algorithms used within the context of product development”, *Procedia CIRP*, Vol. 91, pp. 508–515. https://doi.org/10.1016/j.procir.2020.02.207

Sarang, P. (2023), *Thinking Data Science: A Data Science Practitioner’s Guide*, The Springer Series in Applied Machine Learning, Springer International Publishing. https://doi.org/10.1007/978-3-031-02363-7

Schmid, T., Hildesheim, W., Holoyad, T. (2021), “The AI Methods, Capabilities and Criticality Grid”, *KI - Künstliche Intelligenz*, Vol. 35 No. 3-4, pp. 425–440. https://doi.org/10.1007/s13218-021-00736-4

Sonntag, S., Luttmer, J., Pluhnau, R., & Nagarajah, A. (2023). A PATTERN LANGUAGE APPROACH TO IDENTIFY APPROPRIATE MACHINE LEARNING ALGORITHMS IN THE CONTEXT OF PRODUCT DEVELOPMENT. *Proceedings of the Design Society*, 3, pp. 365-374. https://doi.org/10.1017/pds.2023.37

Waring, J., Lindvall, C. and Umeton, R. (2020), “Automated machine learning: Review of the state-of-the-art and opportunities for healthcare”, *Artificial Intelligence in Medicine*, Vol. 104, p. 101822. https://doi.org/10.1016/j.artmed.2020.101822

Zhang, D., Maslej, N., Brynjolfsson, E., Etchemendy, J., Lyons, T., Manyika, J. (2022), The AI Index 2022 Annual Report, AI Index Steering Committee, Institute for Human-Centered AI, Stanford University, Stanford, CA.