Design Science Benchmarking – A Structured Approach to Make Supervised Machine Learning Algorithms Comparable

BACHELOR THESIS

the Faculty of Business Administration and Economics of the
UNIVERSITY AUGSBURG

for the attainment of the academic degree
„Bachelor of Science“

Universität Augsburg
Prof. Dr. Hans Ulrich Buhl
Chair of Business Administration, Information
Systems, Information & Finance Management

Universität
Augsburg
University

Supervised by:	Prof. Dr. Hans Ulrich Buhl
Submitted by:	Julian Armin Dormehl [redacted]
Submission date:	13.01.2022

Abstract

Current developments, including high availability of data and ever rising computing power, constantly enable new approaches in the field of artificial intelligence. By using algorithms from machine learning, an instance can iteratively learn from data and perform cognitive tasks. As a large amount of machine learning algorithms already exists today and is still increasing rapidly, researchers, data scientists, and machine learning engineers must choose which algorithm to apply and optimize to solve their individual problem. In most cases, the selection of a specific algorithm seems to be highly prediction performance motivated as well as dependent on situational tendencies of the user. For use in non-productive environments, these intentions would initially be sufficient. However, for operational applications these tendencies do not provide satisfactory solutions, as they represent a one-sided perspective and an unstructured approach. Thus, when developing productive applications, no methodological comparison is made between machine learning algorithms, that considers factors of the data basis, the operational view and the explainability of a machine learning model, in addition to the commonly used metrics for evaluating predictive performance. To close this gap we develop two artifacts, first a structured benchmarking procedure for the comparison of supervised machine learning algorithms, a particular paradigm of machine learning we focus on, second a list of criteria, implemented in the benchmarking model, for identifying the most appropriate supervised machine learning algorithm from a holistic perspective following a design science research approach. To provide robust, practical, and user-friendly artifacts, we validate our results in a four-step approach, by, for example, conducting a discussion with research experts, which is prospectively completed by a real-world application of the model. Our results contribute to a structured, generic procedure that supports the benchmarking of supervised machine learning algorithms and provides users with benchmarking-relevant dimensions to identify the most appropriate supervised machine learning algorithm for their individual use-case.

Keywords: Supervised Machine Learning, Algorithm Selection, Multi-criteria Benchmarking, Structured Benchmarking Approach, Design Science Research.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
RQ	Research Question
SML	Supervised Machine Learning
UML	Unsupervised Machine Learning
RL	Reinforcement Learning
SVM	Support Vector Machine
ANN	Artificial Neural Network
CRISP-DM	CRoss Industrial Standard Process for Data Mining
TDSP	Team Data Science Process
KDD	Knowledge Discovery in Databases
IS	Information Systems
TP	True Positives
TN	True Negatives
FN	False Negatives
FP	False Positives
p	Precision
r	Recall
F_β	F-beta score
ROC AUC	Area Under the Receiver Operating Characteristic Curve
RMSE	Root Mean Square Error
MAPE	Mean Absolute Percentage Error
DSR	Design Science Research
DO	Design Objective
AS	Accessibility Score

1. Introduction

Artificial intelligence (AI) is nowadays often seen as a panacea for challenges in various industries and is even attributed the ability to disrupt established business models (Häckel et al. 2021). Since AI applications consistently achieve high performance and increasingly represent a benchmark to other kinds of technologies, far-reaching potentials for solving cognitive tasks arise. Therefore it is no coincidence that AI plays such a prominent role in the so-called Gartner Hype Cycle, which lists technologies that are expected to bring competitive advantages in the coming decade (Gartner 2020). These developments in the field of AI are possible by primarily increasing computing power and increasing connectivity in everyday life which results in a higher amount of data available to be collected (Agrawal et al. 2018). Machine Learning (ML), as today's core of AI, uses algorithms to iteratively learn from this data and perform cognitive tasks. By increasing the use of data, the underlying model gains experience and can increase its prediction performance, which can be used in subsequent process steps (Janiesch et al. 2021). Therefore, ML algorithms are well established, for example in e-commerce or streaming platforms, to make suggestions for future customer activities based on their past activities and search queries (Agrawal and Jain 2017). The classification underlying this use case is one of the core features of supervised ML (SML) – a specific ML paradigm.

Since there already exists a large amount of ML algorithms that produce different results when applied to a dataset, researchers, data scientists, and machine learning engineers must choose between a variety of methods to solve their individual problem (Domingos 2012; Jordan and Mitchell 2015). On the online platform Kaggle, for example, the selection of concrete ML algorithms is carried out through a competition, with the winner being the one who applies an algorithm best tailored to a given use case (Ketter et al. 2016). In this case, as in practical projects, the evaluation often refers to single performance metrics, such as the prediction accuracy of the applied algorithm (Kotsiantis 2007; Zaharia et al. 2018). In general, ML research focuses heavily on improving prediction performance (Wenninger et al. 2022). Therefore, a simultaneous consideration of different functional requirements is often neglected in practice, which consequently has influence on the algorithm selection process (Cawley and Talbot 2010). Thus, ML algorithms are often selected in a relatively ad-hoc manner based on past experience or individual preferences (Domingos 2012; Hill et al. 2016). For non-productive solutions and applications in test environments, the situational selection process is initially sufficient.

However, for productive solutions that are supposed to deliver added value to customers, prevailing practices in the algorithm selection process represent an inappropriate approach. In addition to the metrics used for prediction performance evaluation, further factors exist that influence the overall performance in an operational application (Brodley and Smyth 1995; Janiesch et al. 2021). For example, criteria such as the required computation time and, especially in sensitive operating environments, the energy consumption of ML algorithms represent requirements that can have a negative impact on customer satisfaction – regardless of prediction performance – if they are not adequately met. The explanatory power of a ML model may also be insufficient when it is used in a decision-critical domain, such as health care, where the correctness of the predictions is indispensable. Since ML relies on the initiation of data and algorithms make different predictions based on it, its nature as an upstream factor also influences productive applications. Consequently, current practices represent a one-sided perspective due to their strong focus on prediction performance and lack of consideration of application-relevant factors. This tendency is complemented by a seemingly unstructured approach, as the development of ML solutions requires numerous development cycles. Thus, it turns out to be challenging and problematic at the same time for practitioners to identify most suitable ML algorithms for a particular use case from the multitude of available algorithms.

Therefore, a structured and comprehensive procedure is required to provide guidance to practitioners in performing complex and nested development steps in ML projects. Since different algorithms perform differently according to application-relevant criteria, comparisons of possible ML algorithms should be conducted in view of the goal of high-performance end applications. The methodology of *Benchmarking* – a former management technique – represents a suitable framework for this purpose as it enables a continuous comparison of ML algorithms and a systematic application and therefore optimization of selected methods. Thus, ML algorithms considered for the application – so called benchmarking candidates – are compared with a benchmark that has the best overall performance in the respective use case. After thorough evaluation cycles, the most suitable ML algorithm for a use case can be identified. Furthermore, due to the meta-model character of the *Benchmarking* methodology, it is possible to compare ML algorithms from a holistic perspective using several application-relevant criteria supplementary to the already extensively used prediction performance evaluation. Hence, in addition to the unstructured approach, the one-sided perspective in the evaluation and selection of ML algorithms can be addressed. Since it is our ambition to develop both a generic and an individually applicable model, we focus on algorithms of the supervised learning paradigm. To develop the associated solution approach and to close the mentioned gaps, the following research questions (RQs) are formulated:

> *RQ1: What should a structured procedure look like to be able to compare supervised ML algorithms in the sense of Benchmarking?*
> *RQ2: Which criteria support the structured Benchmarking in identifying the most appropriate supervised ML algorithm from a holistic perspective?*

In this sense, the goal of this thesis is to develop a generic as well as a structured process by which ML methods can be benchmarked (RQ1). In this way it should be possible for the user to compare different methods when following the recommended approach. With the aim of laying foundation of a multi-criteria benchmarking-model, a conceptual model is introduced that goes beyond the already frequently used technical evaluation criteria of prediction performance. On the one hand, the single-criterion-driven perspective is extended to steps upstream and downstream of the ML model, including data basis and explainability, and on the other hand also to operational factors (RQ2). By evaluating the potential from a holistic perspective, the most suitable algorithm for the particular use case should be identified.

Although the underlying problem is partially known in the field of research, this thesis argues improvement to existing solutions along the lines of Gregor and Hevner (2013) for the following reasons: (1) This thesis, in addition to considering predictive performance, is one of the first to highlight the *Benchmarking* of SML algorithms based on multi-criteria dimensions that are important for evaluation and the following application. (2) With the help of a structured procedure for benchmarking SML algorithms, practitioners can be provided with a guide that does not allow the selection of methods according to situational tendencies. Furthermore, the process model documents the step-by-step application of specific algorithms in the context of SML.

This thesis is structured in seven sections. After this introduction, the second section deals with the theoretical background of ML, the approach of *Benchmarking* as well as the motivation for users of the described concept by analyzing existing literature and comparing state-of-the-art processes/concepts. Section 3 describes the methodological approach before introducing the *Benchmarking* approach and the conceptual model for multi-criteria assessment in section 4. Section 5 evaluates both models, which is followed by a comprehensive discussion in section 6 before the thesis concludes in section 7.

2. Theoretical Background

2.1 Artificial Intelligence and Machine Learning

AI is one of the most promising technologies that is continuously evolving, and its development has not yet come to an end. In particular, the availability of data and increasing computational capabilities are responsible for technological progress in this field so far (Berente et al. 2021). As a result, more data can be processed in less time, enabling AI technologies to pursue their goal of mimicking human-like decision-making (Brynjolfsson and Mitchell 2017). Russel and Norvig therefore define AI as the automation of rational behavior and the simulation of human behavior (2016). Despite numerous definitions, there is still no uniformity specifying the term (Collins et al. 2021).

Similar difficulties for identifying a unified definition exist in the field of ML, which is the core of AI today (Berente et al. 2021). In general, ML represents a paradigm for improving the performance of a computer program with respect to a task by gathering problem-specific experience (Jordan and Mitchell 2015). In contrast to conventional approaches, the problem solving is based less on manual programming and more on automated learning processes (Brynjolfsson and Mitchell 2017). Accordingly, available data is an indispensable ingredient for these learning processes to bring about useful insights, decisions, and predictions related to a task (Jordan and Mitchell 2015). Since the output depends significantly on the quality as well as the quantity of available data, ML projects initialize as much task-specific data as possible and process it in further steps (Agrawal et al. 2019; Burkart and Huber 2021; Kessler and Gómez 2020). Within the ML lifecycle, this step is described as *Feature Engineering*. During the process, suitable representations are extracted from the amount of data – so called features – if certain data elements negatively influence each other or are not appropriate for the learning process (Domingos 2012; Janiesch et al. 2021; Kotsiantis 2007). For example, *Feature Engineering* can withhold discriminating information from the subsequent learning process that would otherwise affect the generated output. Conversely, *Feature Engineering* can also be used to highlight data attributes related to a target output (Brynjolfsson and Mitchell 2017; Kessler and Gómez 2020). In a further step, the preprocessed data is used by an individually selected ML algorithm to identify patterns and relationships with respect to the task (Häckel et al. 2021). Thus, an underlying model is developed based on the training data using the iterative manner of the respective algorithm (Janiesch et al. 2021). In the context of this step, the data basis can be thought of as a training set of N elements $\{x_1, \ldots, x_N\}$, based on which an algorithm-specific function $y(x)$ is applied that maps the given input to an output (Bishop 2006). Since the goal of a ML model is to make correct predictions based on unknown examples, training data sets are used to increase the generalization capability of the underlying ML model (Bishop 2006; Döbel et al. 2018; Domingos 2012). After sufficient validation of the ML model in several training cycles, the deployment in a productive environment can take place (Kühl et al. 2021). In terms of explanation, figure 1 provides a rough overview of the ML lifecycle.

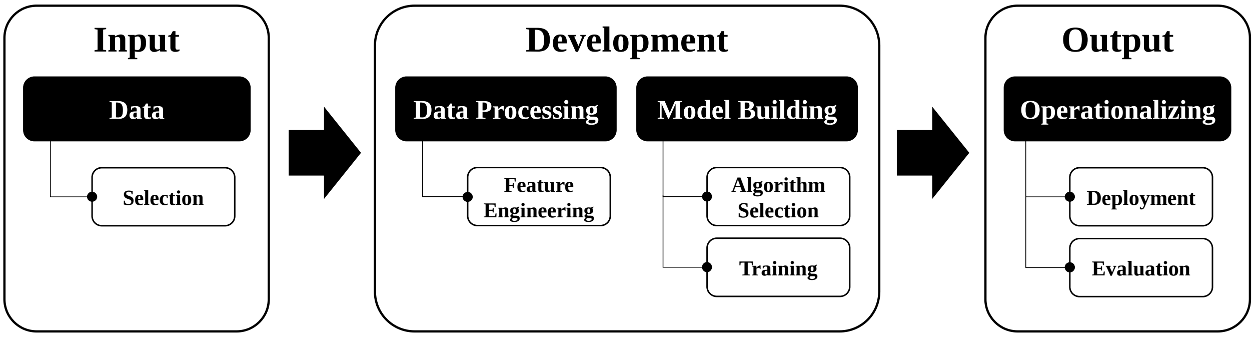

Figure 1: Simplified representation of the ML lifecycle
Own representation derived by underlying literature

With respect to the task and the underlying data basis, three techniques can be distinguished in ML, called supervised ML (SML), unsupervised ML (UML), and reinforcement learning (RL). SML uses samples of data consisting of labeled input-output pairs to train an underlying model that predicts an output given unknown input (Janiesch et al. 2021; Kühl et al. 2021). Depending on the target values, SML can perform two different tasks. For discrete target values, the input values can be assigned to predefined categories in the sense of a classification. In contrast, regressions can be performed using appropriate algorithms to predict continuous target values. (Bishop 2006). An exemplary algorithm for classification is the support vector machine (SVM), which divides values into two classes with the help of hyperplanes, so that the largest possible object-free space remains around the separating plane (Kotsiantis 2007). Artificial neural networks (ANNs) can perform both classification as well as regression tasks by modifying input parameters by weighting contained nodes as part of an iterative process to derive an overall result (Bishop 2006; Döbel et al. 2018). UML as a further ML technique is used in particular when suitable classification groups – elements with common properties – are to be found within unlabeled data, although no suitable definition of the groups is available (Bishop 2006; Janiesch et al. 2021). Accordingly, this technique is often suitable for finding clusters in advance of the application of SML algorithms. In contrast to the other two approaches, RL defines a goal that is to be achieved entirely according to the principle of trial and error as well as under predefined framework conditions. By finding suitable actions, the model receives a reward and is thus conditioned during the learning process regarding behaviors (Janiesch et al. 2021). Since today's ML applications are mostly subject to supervised learning and the techniques differ greatly in terms of learning processes, we deliberately focus on SML algorithms (Jordan and Mitchell 2015; Kotsiantis 2007).

2.2 Practices and Characteristics of Machine Learning Endeavors

ML generally is a data-driven approach and thus represents a central solution for processing large amounts of data. In particular, ML can address the challenges of the 3 V's consisting of volume, velocity, and variety of available data (Kessler and Gómez 2020). Conversely, the application of ML is also highly dependent on the underlying data. Here, arguably one of the biggest challenges for data scientists on ML projects is poor data including missing or even duplicated values (Burkart and Huber 2021; Gudivada et al. 2017). Accordingly, error corrections in the data sets are often necessary to guarantee applicable ML models. Increasingly, ML engineers are also confronted with a large availability of data, which results in more time required to train a ML model (Domingos 2012). Consequently, a trade-off between time to deployment and collecting as much data as possible to achieve powerful ML models is required. Furthermore, depending on the data set, its structure and the weighting of the features, different ML algorithms are suitable. Therefore, their selection should be made depending on the individual use case (Polyzotis et al. 2018). In general, different ML algorithms have individual advantages and disadvantages, depending on the underlying data (Kotsiantis et al. 2006). Accordingly, the dimension of data not only affects the performance of a ML model and its training time, but also influences the selection of a corresponding algorithm. Due to the different and partly opposing perspectives on it, the evaluation of its quality becomes correspondingly complex (Burkart and Huber 2021). Nevertheless, due to the strong dependency of the output of an ML model as well as its learning capability, the nature of the underlying data basis cannot be ignored when selecting an algorithm.

Once the data is processed and specific algorithms are selected for training a ML model, operational factors must be considered. Generally, algorithms differ in terms of their required computation time and associated energy consumption (García-Martín et al. 2019). Depending on the location and energy prices, different computational costs arise. As it becomes necessary in emergence of Green Information Systems (IS) to deliver energy efficient and thus sustainable implementations, energy savings through ML applications compared to the consumption of an alternative computational approach are beneficial (Lehnhoff et al. 2021; Wenninger et al. 2022).

In addition, as already described, the nature of the data basis influences the required computation time. For example, SVMs take longer to compute if the data basis is more extensive (Goodfellow et al. 2016). To remove mutually negative, irrelevant and redundant features from a data set and reduce the computing time of an algorithm, the feature subset selection method represents a suitable solution. (Kotsiantis et al. 2006). Nevertheless, the demand for operational resources remains higher compared to heuristics and engineering methods, which are subject to the logic paradigm. Their methodological framework specifies a step-by-step approach that yields deterministic results. ML algorithms, on the other hand, undertake probabilistic deductions based on underlying data and follow the learning paradigm (Gustavsson and Ljungberg 2021). Therefore, ML algorithms - compared to logic-focused methods – are able to achieve higher prediction performances related to a task (Wenninger and Wiethe 2021). The improvements in predictive performance result from, among other things, a smaller number of constraining measures, easier handling of large data sets, and accounting for interactions between variables (Müller et al. 2016). Despite the high prediction performance that can be achieved using ML algorithms, even relative to other methods, operational factors must not be neglected with respect to training cycles and real-world applications. In the underlying literature analysis, only a few publications deal with this issue.

In the context of ML model training, metrics of the prediction performance are applied in the validation process prior to overall evaluation. These practices are particularly justified by the fact that even the smallest changes, for example in *Feature Engineering*, can affect the performance of a model (Sculley et al. 2014). Since the prediction performance metrics can be made quantitatively measurable, they represent sufficient indicators of the quality of changes made. However, for the final selection of an ML algorithm – after enough training cycles – further factors must be considered as well. Therefore, the predominant focus on prediction performance in the selection of a ML algorithm represents a critical step (Kotsiantis 2007).

To address prediction performance metrics for classifications, a digression is needed for the underlying assumption. Accordingly, we assume binary use cases, where for each data point a class predicted by the classifier must be compared to the actual condition (Kratsch et al. 2021). Therefore, a true positive (TP) classification occurs when an actual positive condition is classified as a positive condition and a true negative (TN) occurs when an actual negative condition is classified accordingly. Misclassifications occur when an actual positive condition is classified as a negative (FN), or actual negative conditions are classified as positive condition (FP) (Häckel et al. 2021). Figure 2 gives a corresponding overview for the predictions of a classifier.

		Actual	
		Positive	Negative
Predicted	Positive	*True Positive (TP)*	*False Positive (FP)*
	Negative	*False Negative (FN)*	*True Negative (TN)*

Figure 2: Classification outcomes in a 2 x 2 confusion matrix
(Agarwal 2019)

Hereby, the theoretical foundation for the prediction performance of a classification is laid. A pre-selection of common prediction performance metrics for SML algorithms and thus classification and regression tasks is listed in table 1.

Selecting a predictive performance metric tailored to the specific use case is essential for a proper evaluation of a ML endeavor. In this context, each performance metric has its own advantages and disadvantages (Kühl et al. 2021). In general, the metric of *Accuracy* is probably the most common metric for evaluating the prediction performance of ML algorithms, as it is usually a reliable indicator (García-Martín et al. 2019; Kotsiantis et al. 2006; Kratsch et al. 2021). However, *Accuracy* may lose meaning in the case of unbalanced data. For example, it has virtually no meaning when the failure of a nuclear power plant must be predicted based on a data set that only include the plant's functionality. Therefore, a focus on multiple metrics is recommended. The use of composed metrics such as the *F-beta score* also represents a feasible solution, as it allows the aggregation of *Precision* and *Recall* into one metric (Kühl et al. 2021).

Task	Metric	Description	Formula
Classification	Accuracy	*Accuracy* is the proportion of true results among the total number of cases examined.	$\frac{(TP+TN)}{(TP+FP+FN+TN)}$
	Precision (p)	*Precision* is the proportion of predicted positive conditions consistent with the actual conditions.	$\frac{TP}{(TP+FP)}$
	Recall (r)	*Recall* is the proportion of all positive conditions consistent with a correct classification.	$\frac{TP}{(TP+FN)}$
	F-beta score (F_β)	The *F-beta score (F_β)* allows to aggregate *Recall* and *Precision* into one metric, using the parameter β to adjust the balance of both. The smaller the beta value, the more weight is given to *Precision*. The reverse is true to *Recall.*	$(1+\beta^2)\times\frac{p\times r}{(\beta^2\times p)+r}$
	ROC AUC	The *Receiver Operating Characteristic curve (ROC)* corresponds to a graph showing the performance of a classification model at various classification thresholds. The *Area Under the ROC Curve (AUC)* indicates the ability of a classifier to avoid false classifications.	Since the *ROC* curve plots the *Recall* (True Positive Rate) against the False Positive Rate at different thresholds and consequently determines the *AUC*, the metric is solved graphically (Appendix A).
Regression	RMSE	The *Root Mean Square Error (RMSE)* measures the quality of predictions by calculating the deviations between predictions $\hat{y}(i)$ and the actual measured values $y(i)$.	$\sqrt{\frac{\sum_{i=1}^{N}(y(i)-\hat{y}(i))^2}{N}}$
	MAPE	The M*ean Absolute Percentage Error (MAPE)* measures the quality of predictions by calculating the mean of the percentage deviation between predicted values $\hat{y}(i)$ and actual values $y(i)$.	$\frac{\sum_{i=1}^{N}\left\|\frac{y(i)-\hat{y}(i)}{y(i)}\right\|}{N}$

Table 1: Selection of frequently used prediction performance metrics
(Agarwal 2019; Zuccarelli 2021)

The output of a ML model can also be evaluated in terms of its comprehensibility by humans. As the degree of complexity and opacity of existing ML models increase, even experts are often unable to fully grasp the interrelationships in ML models (Burkart and Huber 2021; Schaaf et al. 2021). However, the ability of a ML model to explain its results and ensure traceability is a key requirement for productive applications in certain domains (Barredo Arrieta et al. 2020; Burkart and Huber 2021; Biran and Cotton 2017). Thus, in critical contexts such as health care there is a legitimate interest in being able to explain and understand deployed ML models whenever derived predictions can have serious consequences on a human live (Burkart and Huber 2021; Schaaf et al. 2021). Therefore, depending on the use case, an evaluation of the output with respect to the explainability of an underlying ML model is required (Barredo Arrieta et al. 2020). To create transparency in this respect, proprietary methods have been developed in the eXplainable AI (XAI) research area. These methods can be used as an upstream or downstream step of the application of ML models (Rosenfeld 2021; Zhou et al. 2021). Along with the application of XAI methods, the output factors of a ML model can also be evaluated with respect to the degree of explainability, which is mainly based on subjective and individual judgements. In general, the occurrences of explainability and prediction performance represent a trade-off since complex ML algorithms can achieve high prediction performance while providing little insight into the process of making predictions. This is exemplified by ANNs as black-box models whose processes leading to the final prediction cannot be understood in detail due to their large variable space. On the other hand, ANNs are among the most powerful algorithms with respect to their prediction performance and robustness in practical applications, illustrating the prevailing trade-off (Barredo Arrieta et al. 2020; Mohseni et al. 2021). Accordingly, it is necessary to consider the requirements for explainability and prediction performance in advance of the application of a ML model in productive environments (Burkart and Huber 2021). In general, however, further factors, such as operational aspects or the nature of the data basis, should be considered along with such deliberations. By considering all the enumerated dimensions, decisions regarding the development of productive end applications can be made from a holistic as well as objective perspective.

Overall, the ML lifecycle practices prevalent in each step have a strong impact on the output of an ML model, which in turn can be viewed from different perspectives and evaluated accordingly. From a broader perspective, the individual steps are highly nested and, especially in the development process of a ML model, dependent on the user’s situational tendencies, which makes it difficult to repeatedly apply predefined processes to different situations (Kühl et al. 2021). Currently, there exist no explicitly tailored approaches for structuring projects in the field of ML. Accordingly, approaches from other fields are often used, such as the CRoss Industry Standard Process for Data Mining (CRISP-DM) (Kessler and Gómez 2020; Studer et al. 2021). CRISP-DM is a generic process model that is used for planning, communication, and documentation within and outside the project team. The process model follows the generic steps Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment and is commonly applied to ML efforts as an established industry standard (Wirth and Hipp 2000). However, CRISP-DM does not ensure predefined quality levels and therefore lacks guidance on quality assurance, as the procedure does not specify criteria that can be evaluated from a management perspective (Wirth and Hipp 2000; Studer et al. 2021). The Team Data Science Process (TDSP) represents an approach, which is a flexible, iterative data science methodology to deliver so-called intelligent applications. In general, the TDSP approach has great similarities with CRISP-DM but focuses even more on the area of data science including the modeling of data (Microsoft 2021). Complementary to this, the Knowledge Discovery in Databases (KDD) approach can be used to discover relevant patterns in data structures (Fayyad et al. 1996). Both TDSP and KDD can be used as part of ML efforts to generate feasible solutions, but do not represent a structured process across the entire ML lifecycle, instead focusing specifically on data science tasks and extracting knowledge from data. Compared to TDSP and KDD, the CRISP-DM approach is more holistically applicable to ML projects. However, due to its data mining focus, it neglects the iterative application of the model on unknown data and the model building process (Studer et al. 2021). To fulfill the requirements of considering the ML lifecycle and granting quality assurance standards, which could not be sufficiently addressed by prevailing approaches, the methodology of *Benchmarking* is introduced in the following section as a possible solution approach.

2.3 Benchmarking

Benchmarking is a well-known management technique that supports companies in identifying best practices to achieve better performances. The methodology became known through a use case at XEROX Corp. where *Benchmarking* was applied to reduce manufacturing costs to remain competitive. Therefore, efforts have been made to understand what competitors do better with the aim of adapting their value-creating practices. This way, competitiveness could be maintained (Harvard Business Review 1987). From this use case, *Benchmarking* has evolved into an approach that pursues continuous measures and comparisons with the goal of identifying better practices and, consequently, improving overall performance by applying them accordingly (Teuteberg et al. 2009; Watson 1993). As *Benchmarking* also leads to faster organizational learning, it can also be associated with business process redesign or other change initiatives (Drew 1997).

Although the applications of *Benchmarking* vary widely today, the core of *Benchmarking* remains. Comparisons are still made based on individually selected performance metrics, and the comparisons are made to standards or targets, a so-called benchmark. Accordingly, in benchmarking ML algorithms, separate tests are performed on the same data basis and compared with each other. The first tested ML algorithm is therefore a benchmark for the following ML algorithms as long as it shows a higher overall performance than the following ML algorithms (Rautu et al. 2017).

Since the methodology of *Benchmarking* represents a meta-model that drives best practices, it represents an adequate approach in the context of ML to address the needs that are insufficiently addressed by CRISP-DM, TDSP and KDD. First, since no standard process exists for the holistic execution of ML projects, the benchmarking approach can integrate the ML lifecycle to guarantee a structured approach to ML endeavors. Second, *Benchmarking* allows the integration of application-relevant criteria and thereby a structured comparison of the best performing available algorithms with respect to predefined criteria for a use case. Accordingly, quality assurance standards can be formulated, which can be constantly re-evaluated and adapted as part of the iterative procedure. Therefore, ML endeavors get less dependent on individuals or teams involved, as a structure is given, which facilitates documentation of ML projects, and allows successful practices to be repeated more frequently. Thus, our research contributes to the development of a (process) model that nowadays belongs to the less popular contributions of AI studies in IS research and accordingly offers large scope for open research (Collins et al. 2021).

3. Methodology

To enable the development of a generic, structured process for benchmarking SML algorithms, we follow the Design Science Research (DSR) paradigm. The DSR methodology was chosen as it provides a structure for users to gain an understanding of problems related to the use case at hand, while building a bridge to application-oriented solutions. As part of the process, artifacts are developed that contribute to domain-specific knowledge based on the abstraction of reality (Gregor and Hevner 2013; Hevner et al. 2004). Accordingly, DSR is considered as a problem-solving paradigm that serves human purposes (Hevner et al. 2004; March and Smith 1995). In addition, DSR represents a systematic approach that allows the incorporation of separate methods such as, in our case, an in-depth evaluation of developed artifacts (Gregor and Hevner 2013; Häckel et al. 2021). Thus, within the framework itself, we can create a structured procedure by systematic advice. In general, our intentions and the methodological framework fit well together, as both approaches aim to ensure a strong application focus and a generic character of the artifact. We follow an established, iterative DSR process, which includes five phases as illustrated in the following figure.

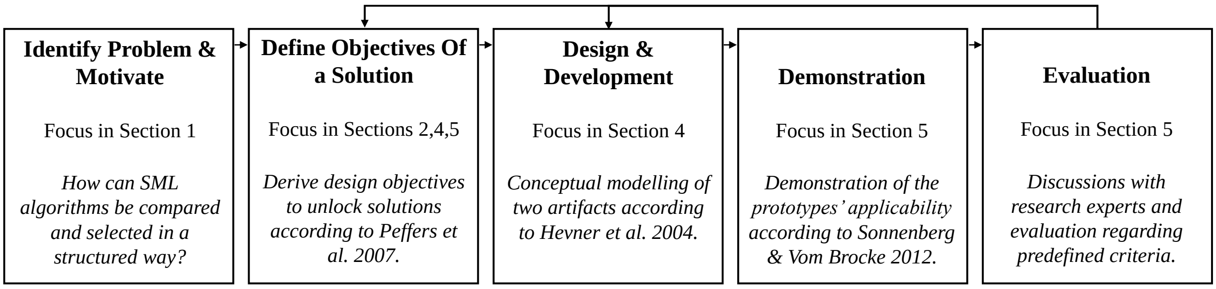
Identify Problem & Motivate
Focus in Section 1
How can SML algorithms be compared and selected in a structured way?
Define Objectives Of a Solution
Focus in Sections 2,4,5
Derive design objectives to unlock solutions according to Peffers et al. 2007.
Design & Development
Focus in Section 4
Conceptual modelling of two artifacts according to Hevner et al. 2004.
Demonstration
Focus in Section 5
Demonstration of the prototypes' applicability according to Sonnenberg & Vom Brocke 2012.
Evaluation
Focus in Section 5
Discussions with research experts and evaluation regarding predefined criteria.

Figure 3: Design Science Research process
(Peffers et al. 2007)

In the first phase, we identify the research problem and justify the value of the proposed artifact. As mentioned in the introduction, there are a variety of SML algorithms whose selection is usually not subject to a structured process, allowing for one-dimensional selection perspectives and situational approaches. However, an optimal selection tailored to the application is rarely made in this way. Practice and research therefore need a structured and comprehensive approach to make SML algorithms comparable with each other in the sense of *Benchmarking*. To identify most suitable algorithms for a use case, application-relevant criteria must be considered as part of the procedure. In the theoretical background, we also go into more detail about the underlying problem to lay foundation for the formulation of design objectives (DOs). These DOs guide the development of the model as a solution to the stated RQs. In the design and development phase, we produce two viable artifacts following the guidelines of Hevner et al. for executing DSR (2004). Subsequently, the artifacts are evaluated to determine its value based on criteria such as *Validity*, *Utility*, *Quality*, and *Efficacy*. In addition, we continuously demonstrate and evaluate our model during the development process to ensure the applicability of our artifacts using the framework proposed by Sonnenberg and Vom Brocke (2012). The framework allows us an in-depth evaluation of our artifacts according to four established steps (EVAL1 – EVAL4), each including ex-ante and ex-post perspectives. As part of the EVAL1-step, a semi-structured literature review was conducted as well as DOs were defined presented in Section 4, ensuring that the model represents a solution to a meaningful DSR problem with relevance for practitioners. In the second step, existing approaches such as CRISP-DM, TDSP and KDD are compared with the ideas of our model in terms of *Benchmarking* (EVAL2). This allows us to validate our DOs and to show that an artifact design provides the solution to the stated problem. In the further procedure, the model artifact was developed considering an ex-ante/ex-post perspective, whereby evaluations are made before and after the artifact instantiation, respectively. To draw initial inferences about our developed artifacts from a theoretical perspective, the designed artifacts were presented to a group of researchers from the field of ML and digital value networks. Domain-specific feedback can be obtained in a subsequent discussion, and the issues raised will be considered in the current drafts of the corresponding artifacts (EVAL3). This thesis provides feasible solutions up to the evaluation of applicable prototypes. To ensure that our artifacts can be applied to real-world challenges, the artifacts will be subjected to a real-world application in perspective (EVAL4). Thereby, different SML algorithms will be tried on a single data set to identify the best SML algorithm for a given use case based on our approach. In summary, our research contributes to what Gregor and Hevner call *improvement* of solutions that exist in practice and research (2013). We argue that the benchmarking artifacts developed represent far more domain-specific solutions to prevailing problems and thus new solutions for known problems.

4. Results

In this section, we present two model artifacts developed through the DSR methodology. With the help of our model artifacts, users should be able to identify the most suitable SML algorithm for their individual use case. Since we want to enable both a structured approach and a holistic consideration of application-relevant criteria, we draw on the methodology of *Benchmarking*. This is intended to replace the prevailing insufficient practices, identify best SML algorithms, and generate powerful productive applications. This section is structured into the definition of DOs (Section 4.1), an introduction to the benchmarking model (Section 4.2) and the listing of application relevant criteria (Section 4.3).

4.1 Design Objectives

To guide the development and evaluation of our artifacts, we derived three DOs from the prevailing problem setting, spanning up a solution space. In the following, the DOs are introduced and justified, each in relation to the implementation in the artifacts.

As discussed in section 2.1, there are major differences between ML techniques in terms of learning processes and specific outcomes. While SML develops its underlying model based on labeled data, an UML approach is tasked with recognizing patterns without pre-existing labels. By learning from its environment, the RL technique is completely detached from the other two approaches. Compared to SML, RL and UML techniques are evaluated with entirely different metrics because of the differences in the tasks they perform. RL approaches also have a unique model design and development process due to integrated rewards. To create generic as well as specific applicable artifacts we focus on one specific ML technique. As we want to have maximum impact on prevailing practices and SML represents the most used ML technique today, we focus the development of our artifacts on SML algorithms.

DO.1. *The benchmarking artifacts shall exclusively consider SML algorithms to meet requirements of a generic but at the same time specific applicable procedure.*

As discussed in Section 2.2, *Benchmarking* has a wide range of applications today, and its use is often driven by several purposes and motivations. To consider steps upstream of benchmarking SML algorithms in our artifacts, motivations shall be considered as triggers that initiate the benchmarking process. Accordingly, for each of the benchmarking steps there are associated motivations whose occurrence can serve as an entry point. This is to ensure that the bridge is built between preliminary considerations made to practical implementations.

DO.2. *The benchmarking procedure should consider the steps upstream of its own process, which includes motivation as an entry point to ensure relevance to the application.*

Since it is a prevailing practice in the selection process of ML algorithms to select exclusively according to prediction performance, adverse tendencies become widespread. Thus, ML algorithms are partially selected ad-hoc based-on experience. To identify the most suitable ML algorithm for a use case independent of exclusive consideration of the prediction performance, a solution approach is to be developed. The methodology of *Benchmarking* represents a suitable approach for this purpose, as it corresponds to a structured procedure that can compare most feasible solutions by nature. By integrating application-relevant criteria that are specifically related to SML applications, the identification of best performing SML algorithms can be considered from a holistic perspective.

DO.3. *The approach of our artifacts is to follow a structured procedure in the sense of Benchmarking, drawing on criteria relevant for the comparison of SML algorithms.*

By following the three DOs, we aim to provide an applicable benchmarking procedure that makes SML algorithms comparable to each other. Our model enables users to build and validate a ML model in a structured way according to the ML lifecycle presented in section 2.1. Due to the nature of *Benchmarking*, most suitable SML algorithms can be identified for an individual use case. Since the benchmarking model addresses different research questions, two artifacts are developed. Following a top-down approach, the framework in terms of *Benchmarking* is presented first (RQ1), followed by criteria relevant for the benchmarking of SML algorithms (RQ2).

4.2 Benchmarking Model

The benchmarking model addresses RQ1 and thus the question how a structured procedure should look like to compare SML algorithms in terms of *Benchmarking*. Figure 4 shows the corresponding artifact considering the derived DOs.

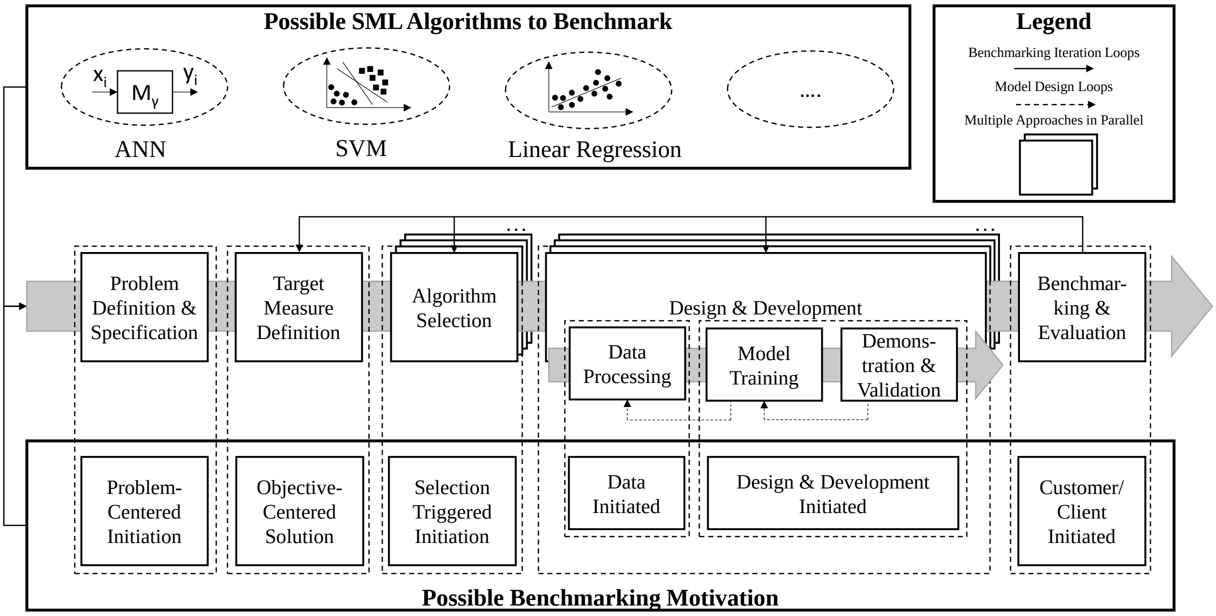
Possible SML Algorithms to Benchmark
x_i
M_γ
y_i
ANN
SVM
Linear Regression
....
Legend
Benchmarking Iteration Loops
Model Design Loops
Multiple Approaches in Parallel
Problem Definition & Specification
Target Measure Definition
Algorithm Selection
Design & Development
Data Processing
Model Training
Demons-tration & Validation
Benchmar-king & Evaluation
Problem-Centered Initiation
Objective-Centered Solution
Selection Triggered Initiation
Data Initiated
Design & Development Initiated
Customer/ Client Initiated
Possible Benchmarking Motivation

Figure 4: Conceptual Benchmarking Model
Own representation

Since we follow a chronological approach in describing our model, we first address a possible benchmarking motivation, an upstream step and thus entry point for our process. By considering motivations in our procedure, a bridge can be built between initial considerations and their practical implementation. We propose six possible entry points in the benchmarking procedure. The first entry point represents the so-called *Problem-Centered Initiation*, which motivates benchmarking from a status quo and a possible problem perspective. Benchmarking of SML algorithms, on the other hand, can also be motivated by a possibility perspective or a goal-oriented view according to an *Objective-Centered Solution*. Both motivations thus represent opposing perspectives to benchmark SML algorithms. The so-called *Selection Triggered Initiation* represents a core motivation in the artifact, as the multitude of SML algorithms and the identification of the most suitable approach in each use case is a challenge that we want to solve. As noted in the description of the ML lifecycle (Section 2.1), the development of a ML model is based on the nature of underlying data and systematic training. Since we want to integrate the ML lifecycle in *Benchmarking* and thus generally motivate ML projects to follow the benchmarking model, we divide possible entry points with respect to design and development cycles into *Data Initiation* and *Design & Development Initiated*. Within this framework, *Feature Engineering* processes and subsequent application-oriented model development cycles can be motivated. Finally, benchmarking processes in general as well as specifically in the case of the comparison of SML algorithms can be motivated by improvement requests or the desire for adjustments on the part of a customer. Therefore, the entry point *Customer/Client Initiated* is exemplary for benchmarking motivations brought about by third parties.

The motivations presented can be individual for each project and take on different characteristics. In addition, the model represents possible suggestions for generic motivations that can provide an impetus for benchmarking SML algorithms. Therefore, there is no universally valid claim to completeness regarding the listed motivations.

Once users are triggered by motivations to carry out respective projects according to the benchmarking model, a structured procedure – the core of our model – comes into play. In contrast to the preceding entry points, the steps included represent a coordinated chain of effects. First, in the context of the step *Problem Definition & Specification* status quo problems are discussed in more detail. By concretizing existing application problems, practitioners should identify possible problem areas of present situations. In a second step, practitioners should define specific goals for the project outcome based on the status quo. Accordingly, the step is called *Target Measure Definition*. The defined goals represent conditions whose fulfillment is to be achieved. After the model has been run successfully, the result is to be measured against the defined technical or operational requirements. The goals set are thereby dependent on the time required to fulfill them. After the specification of underlying problems and the definition of goals regarding the outcome have set a framework for action, practitioners select algorithms according to individual preferences in the *Algorithm Selection* step. Since many different SML algorithms can be used as output for the productive solutions to achieve the set goals, several algorithms can be selected in this step. The set of selected algorithms in the model is illustrated by stacking them into a third dimension. To compare the selected algorithms in terms of their overall performance and suitability for use in productive solutions, they undergo the *Design & Development* step. Accordingly, *Data Processing* is performed for each selected algorithm to adapt the underlying data basis to the SML algorithm. In concrete terms, *Feature Engineering* is carried out based on selected and cleaned data. The framework of the data used remains the same in this step to ensure comparability. In addition, the data basis on which the *Design & Development* process is based reflects at best the real environment in which the use of an SML model is required. This is to ensure that the algorithms are aligned as closely as possible to the later application. Thereby enough data should be generated. The following *Model Training* can be performed by applying the selected SML algorithms. This is to develop a SML model that is adapted to the use case and environment. Using test data that is still unknown to the SML models, the *Demonstration & Validation* step is performed. In this process, new insights can be gained with respect to pending adaption steps, which can be implemented with the help of re-adjustments to the data basis and corresponding new training cycles. Since several re-adjustments are often necessary, the model is iteratively fitted according to so-called *Model Design Loops*. This is to advance the learning process of the underlying model so that trained models can fulfill their full potential with respect to their task. Once the results achieved in the *Design & Development* step are sufficient for respective algorithms from the practitioner's point of view, the *Benchmarking & Evaluation* step follows. In this step, SML algorithms are evaluated with respect to their overall performance and get selected in the sense of *Benchmarking*. The following section provides an overview of dimensions and associated criteria that can be used to evaluate SML algorithms. The SML algorithm that was initially identified as the best-performing serves as a benchmark for evaluating other SML algorithms until a better SML algorithm is identified. After the structured comparison has been completed, a statement can be made about the SML algorithm best suited to a use case. If the identified benchmark does not match the targets defined in the *Target Measure Definition* step, iteration cycles outside the *Design & Development* step, so-called *Benchmarking Iteration Loops*, are required. This gives the practitioner the opportunity to adjust the defined targets based on accumulated experience, to select new algorithms or to invest further effort in the development of the model. The iterative process is carried out systematically until the most feasible solution is identified through *Benchmarking* and defined targets are met.

4.3 Model-integrated Benchmarking Criteria

The need for an overview of application-relevant criteria for our benchmarking model is twofold. In the *Target Measure Definition* step, individual targets must be defined regarding the degree of fulfillment of use case specific criteria. In addition, SML algorithms must be comparable with respect to predefined dimensions in the *Benchmarking & Evaluation* step. By using the predefined dimensions and criteria in the context of target definition and benchmarking, SML algorithms can be selected from a holistic perspective. Practitioners should therefore be enabled to select the criteria that are important for their use case as if from a toolbox and to weight them individually for their use case. As the benchmarking model represents a meta-model, the benchmarking-relevant dimensions can be integrated separately. Figure 5 represents the benchmarking model integrated with benchmarking relevant criteria.

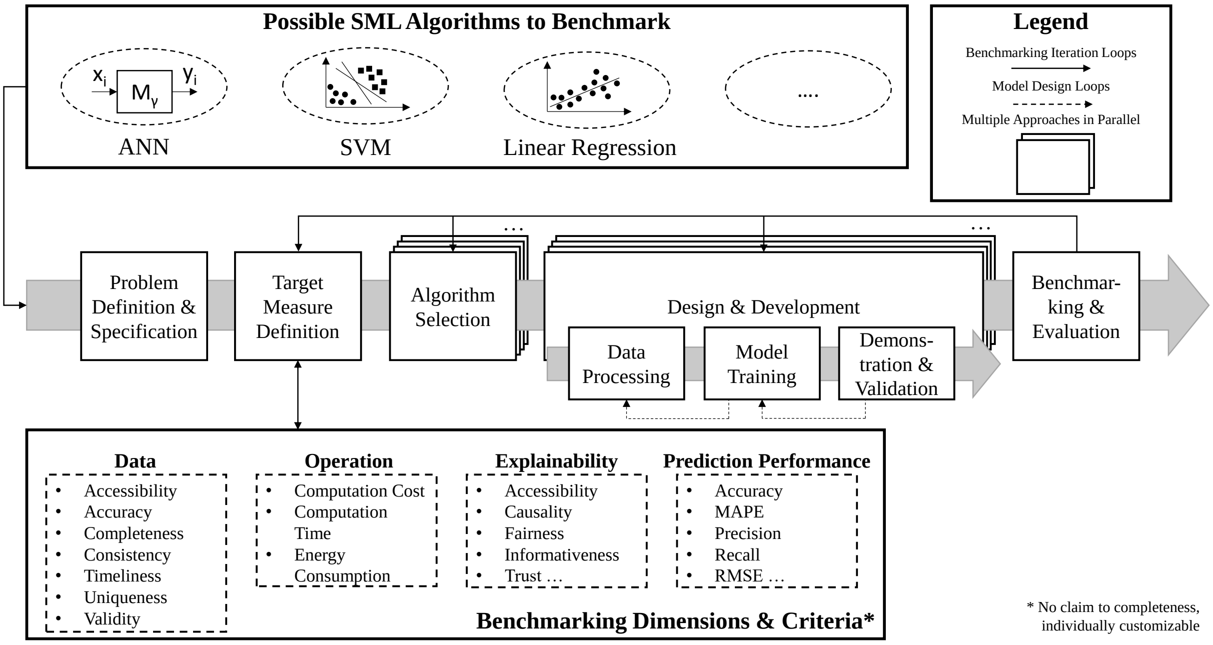

Figure 5: Model-integrated Benchmarking Dimensions and Criteria
Own representation

The benchmarking-relevant dimensions are divided into *Data*, *Operation*, *Explainability* and *Prediction Performance*. The respective dimensions have already been discussed in detail in the theoretical background (Section 2). Each of the four dimensions has measurable criteria that can describe the characteristics of a dimension in an individual use case. All dimensions including their respective criteria are described in a comprehensive listing below.

4.3.1 Data

As discussed in the theoretical background, the nature of the underlying *Data* has far-reaching implications for the application of ML. For example, insufficient *Data* quality negatively affects the predictive performance of a ML model (Burkart and Huber 2021; Kessler and Gómez 2020). Similarly, ML algorithms react differently to the characteristics of its input (Kotsiantis et al. 2006; Polyzotis et al. 2018). To consider the influence of *Data* on the prediction performance on the one hand and the suitability of algorithms with respect to individual use cases on the other hand, *Data* is listed in a separate dimension. Since *Data* may generally differ in essential characteristics, separate criteria are required for in-depth descriptions of *Data* sets (Batini et al. 2009). Thus, the criteria listed in table 2 can provide information on whether the *Data* basis meets requirements for the implementation of ML projects. In general, the criteria can also be represented as parameters, that can be used to compare different *Data* sets in terms of their properties.

	Criterion	Description	Measurement
Data	Accessibility	The *Accessibility* comprises the degree to which objectively correctly initialized parameters in a data set can be grasped by non-experts (Wederhake et al. forthcoming).	Measurement by an *Accessibility Score* (AS) that indicates the extent to which parameters can be captured in a range of [0, 100]. The higher the scores, the better the *Accessibility* is (Wederhake et al. forthcoming).
	Accuracy	The *Accuracy* (acc) of data describes the degree to which it correctly describes an underlying use case (Askham et al. 2013; Batini et al. 2009; Burkart and Huber 2021).	Measurement of the number of data correctly describing an use case (c) in relation to the total number of collected data (d_{total}) (Askham et al. 2013). $acc = \frac{c}{d_{total}}$
	Completeness	The *Completeness* (com) of data comprises the proportion of data collected compared to the data set that could have been collected to describe the underlying use case (Batini et al. 2009; Burkart and Huber 2021).	Measurement of the presence of non-blank values (nb) in relation to the total number of blank values (b) and non-blank values (Askham et al. 2013). $com = \frac{nb}{b + nb}$
	Consistency	The *Consistency* of data describes the degree of correspondence of the semantics of the data to its definition (Batini et al. 2009).	Measurement of data patterns and value frequency over multiple data sets (Askham et al. 2013).
	Timeliness	The *Timeliness* (tim) of data describes the amount of time between collection and use of the data (Batini et al. 2009; Burkart and Huber 2021).	Determination of the time difference between collection (c) and use of the data (u) (Askham et al. 2013). $tim\ = \vert c - u \vert$
	Uniqueness	The *Uniqueness* (uni) of data describes the degree to which duplicates occur in a dataset (Batini et al. 2009).	Measuring the number of records in the real world (r) relative to the number of records in the data set (rd) (Askham et al. 2013). $uni = \frac{r}{rd}$
	Validity	The *Validity* (val) of data describes the degree of correspondence of the syntax of the data with its definition (Askham et al. 2013; Burkart and Huber 2021).	Measurement of the number of valid data (vd) in relation to the total number of data (Askham et al. 2013). $val = \frac{vd}{d_{total}}$

Table 2: Overview of *Data* relevant criteria
Own representation derived by cited publications

4.3.2 Operation

As ML systems increasingly rely on elaborate computer architectures that certainly require high computational power, the consideration of *Operational Factors* in the context of training ML models and applying them in productive solutions become increasingly important (Jordan and Mitchell 2015). Especially in the training phase of a ML model, large amounts of computational time are invested to pursue systematic validation of achieved performances to generate most feasible solutions (Cawley and Talbot 2010). Considering climate goals and the related need for higher energy efficiency, *Operational Factors* are incorporated in our model. To objectively compare SML algorithms with respect to *Operational Factors*, similarly powerful hardware should be used in the benchmarking process. The criteria listed in table 3 are strongly interrelated. For example, the *Computation Time* has a strong influence on the resulting *Energy Consumption* and the resulting *Computation Cost*. However, to enable dedicated comparisons and optimizations in practice regarding the characteristics of *Operational Factors,* the criteria are listed individually.

	Criterion	Description	Measurement
Operation	Computation Time	The *Computation Time* describes the length of time required to perform a computational process.	Measurement of the computation time of an algorithm to perform a predefined task.
	Energy Consumption	The *Energy Consumption* describes the amount of power used to perform a computational task (García-Martín et al. 2019).	Measuring by embedding sensors in the individual components of a system (Wenninger et al. 2022).
	Computation Cost	The *Computation Cost* describes the allocation of costs to the respective consumption of energy units.	Calculation from multiplying cost per unit of energy consumed by total energy consumption.

Table 3: Overview of *Operation* relevant criteria
Own representation derived by cited publications

4.3.3 Explainability

A critical milestone for the widespread application of ML algorithms in productive applications in general is the traceability of their results (Biran and Cotton 2017). Since our artifacts have a generic claim and the comprehensibility of the prediction results is highly relevant in certain application domains, the *Explainability* is considered as a separate dimension. Using the criteria listed in table 4, ML algorithms can be evaluated with respect to their degree of *Explainability*. However, due to the lack of suitable metrics and the prevailing subjectivity in the evaluation as to whether the result is traceable and comprehensible, *Explainability* is much more difficult to measure than other dimensions (Barredo Arrieta et al. 2020; Burkart and Huber 2021). Thus, with respect to the measurement of explanatory power, we address some proposals that may be overtaken by new approaches in the future in a rapidly evolving field of research.

	Criterion	Description	Measurement
Explainability	Accessibility	The A*ccessibility* includes the degree to which end users can make improvements and developments to a given ML model (Barredo Arrieta et al. 2020).	Measurement by using an AS, which indicates the degree of ability to modify model parameters in a range of [0, 100]. The more parameters can be adjusted in relation to a total number of available parameters to be defined, the more accessible an ML model ought to be.
	Trust	The criterion of *Trust* implies an awareness of the behavior and the strengths and weaknesses of the underlying predictive model (Barredo Arrieta et al. 2020; Burkart and Huber 2021).	Measurement by using specific questionnaires (Appendix B) and interviews before deployment (Hoffman et al. 2018; Mohseni et al. 2021; Zhou et al. 2021).
	Causality	The *Causality* of a predictive model describes an understanding of input-output relationships of a model and the relation between data in terms of their attributes and consequent predictions (Barredo Arrieta et al. 2020; Burkart and Huber 2021).	Measurement by using approaches of feature selection (Guyon and Elisseeff 2003), variable importance (Breiman 2001) and in-depth analysis (Yu and Liu 2004). The clearer it is which variable leads to which result, the more causal the result should be
	Transferability	The *Transferability* represents the maturity of a predictive model (e.g. in predictive accuracy), the extent to which it can be applied to as yet unseen task-specific data and entrusted with decision support (Barredo Arrieta et al. 2020; Burkart and Huber 2021).	Measurement by using test data sets to evaluate the trained ML Model, for example, with respect to prediction accuracy. The smaller the difference in prediction accuracy compared to model training, the higher the generalization ability and thus the *Transferability* of a ML model (Roelofs 2019).
	Informativeness	The *Informativeness* of a predictive model describes to what extent internal and decision-relevant information about the problem to be solved is provided (Barredo Arrieta et al. 2020).	Measurement by using specific questionnaires (Appendix C). (Hoffman et al. 2018; Li et al. 2020).
	Fairness	The *Fairness* of a predictive model includes the understandable presentation of results as well as their compliance with ethical standards (Barredo Arrieta et al. 2020; Burkart and Huber 2021).	Measurement of the degree of bias present in the model (Hardt et al.; Speicher et al.).
	Proxy functionality	The *Proxy Functionality* represents further criteria of *Explainability*, that may be indispensable for industry-specific applications (e.g. interactivity, privacy awareness) (Burkart and Huber 2021).	Measurement according to individual approaches.

Table 4: Overview of *Explainability* relevant criteria
Own representation derived by cited publications

4.3.4 Prediction Performance

The metrics for evaluating *Prediction Performance* can be made quantitatively measurable with much less effort compared to the dimension of *Explainability*. Therefore, they often represent the core of the evaluation of ML algorithms today. In general, the achievement of a high *Prediction Performance* is advantageous for most use cases. However, since further dimensions, such as *Explainability*, often behave contrary to *Prediction Performance*, the case may arise that users have to make individual trade-offs with respect to the underlying use case (Barredo Arrieta et al. 2020). When applying the performance metrics, the case can arise where a SML algorithm performs well on one metric but underperforms on another. Accordingly, it is important to evaluate algorithms using different *Prediction Performance* metrics to determine the quality of the underlying model (Caruana and Niculescu-Mizil 2006). Since the individual metrics have already been described in the theoretical background (section 2.2), the following table only serves as a supplement to fully understand the scores achieved by the metrics.

	Task	Criterion	Measurement	Value range	Best value
Prediction Performance	Classification	Accuracy	$\frac{(TP+TN)}{(TP+FP+FN+TN)}$	$[0;1]$	1
		Precision (p)	$\frac{TP}{(TP+FP)}$	$[0;1]$	1
		Recall (r)	$\frac{TP}{(TP+FN)}$	$[0;1]$	1
		F-beta score (F_β)	$(1+\beta^2)\times\frac{p\times r}{(\beta^2\times p)+r}$	$[0;1]$	1
		ROC AUC	Graphical solution (Appendix A)	$[0.5;1]$	1
	Regression	RMSE	$\sqrt{\frac{\sum_{i=1}^{N}(y(i)-\hat{y}(i))^2}{N}}$	$[0;\infty[$	0
		MAPE	$\frac{\sum_{i=1}^{N}\left\|\frac{y(i)-\hat{y}(i)}{y(i)}\right\|}{N}$	$[0;1]$	0

Table 5: Overview of *Prediction Performance* relevant criteria
Own representation derived by Agarwal 2019, Kratsch et al. 2021, Zuccarelli 2021

5. Evaluation

The developed artifacts represent feasible solutions up to and including the EVAL3 step. The results are based on a semi-structured literature review, an in-depth analysis of existing process models and expert discussions. Consequently, to be able to demonstrate and evaluate the application of the developed artifacts in the context of the thesis, an exemplary use case is outlined. Based on this, statements can be made regarding the *Validity*, *Utility*, *Quality*, and *Efficacy* of the artifacts (Gregor and Hevner 2013). To consider a challenging and, consequently, forward-looking topic, the artifacts are evaluated based on a use case of anomaly detection in energy consumption.

Nowadays, the detection of anomalies in energy consumption is an important approach to introduce energy efficiency measures and reduce CO_2-emissions. By predicting the energy demand of a consumer instance, anomalies, deviations from expected conditions, can occur. By detecting specific anomalies resulting, for example, from the use of energy-inefficient equipment, appropriate measures can be taken to counteract them. This can rebalance energy supply and effective consumption of energy. Manufacturing companies in particular benefit from energy anomaly detection, as they generally have a high energy consumption and cost-intensive production cycles (Kaymakci et al. 2021).

To apply our artifacts to a concrete use case, an exemplary manufacturing company under the pseudonym *Production Inc.* is introduced. The company can generate real-time data related to the energy flows prevailing in its production due to integrated smart meters. As environmental restrictions regarding CO_2-emissions arise and large costs are incurred on energy procurement, appropriate measures are to be taken. Due to the meaningful data basis, the application of SML algorithms is suitable to predict the occurrence of energy anomalies and to initiate appropriate measures based on this prediction. To perform SML tasks, labels of the underlying data are necessary. Therefore, the occurring energy anomalies are listed as individual label classes and assigned to the individual data elements of the training data set. Accordingly, data elements with an anomaly are labeled of a respective anomaly class. Data elements without anomaly are set as functional (Kaymakci et al. 2021). Based on this, a corresponding SML algorithm can learn the classification of the data elements to make predictions regarding future anomaly occurrences. Since it is the goal to develop a productive application for energy anomaly prediction, the most suitable SML algorithm should be selected. By using the benchmarking model, a structured approach can be followed during development. In addition, a methodological framework will be given on how *Production Inc.* can meet its objective of predicting anomalies in energy consumption with the best overall performance. Figure 6 provides an overview of the use case specific design of the individual steps in the benchmarking model.

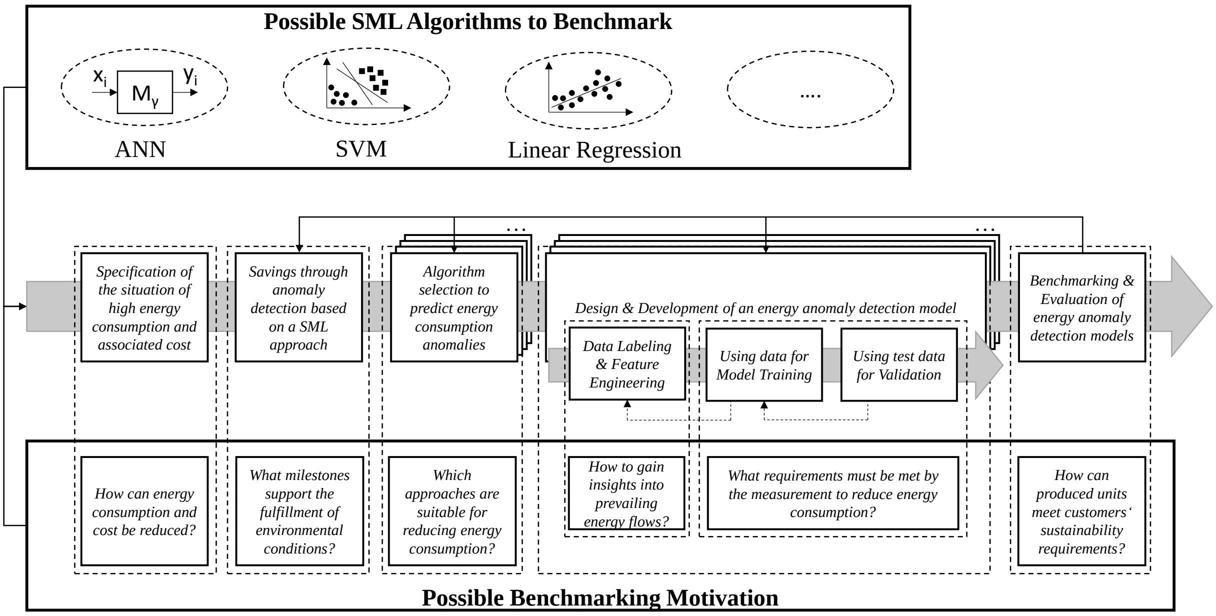
Possible SML Algorithms to Benchmark
x_i
M_γ
y_i
ANN
SVM
Linear Regression
....
Specification of the situation of high energy consumption and associated cost
Savings through anomaly detection based on a SML approach
Algorithm selection to predict energy consumption anomalies
Design & Development of an energy anomaly detection model
Data Labeling & Feature Engineering
Using data for Model Training
Using test data for Validation
Benchmarking & Evaluation of energy anomaly detection models
How can energy consumption and cost be reduced?
What milestones support the fulfillment of environmental conditions?
Which approaches are suitable for reducing energy consumption?
How to gain insights into prevailing energy flows?
What requirements must be met by the measurement to reduce energy consumption?
How can produced units meet customers' sustainability requirements?
Possible Benchmarking Motivation

Figure 6: Benchmarking Model according to the energy anomaly detection use case
Own representation

For the formulation of goals and the holistic evaluation of selected SML algorithms, the overview on benchmarking-relevant dimensions and criteria is utilized. Since the use case of *Production Inc.* is specific and thus individual for the generic overview of the listed dimensions, there are differences in the degree of importance of individual criteria to be considered in benchmarking. Therefore, in figure 7, the criteria that are less relevant for the use case are grayed out.

Data

- Accessibility
- Accuracy
- Completeness
- Consistency
- Timeliness
- Uniqueness
- Validity

Operation

- Computation Cost
- Computation Time
- Energy Consumption

Explainability

- Accessibility
- Causality
- Fairness
- Informativeness
- Proxy Functionality
- Transferability
- Trust

Prediction Performance

- Accuracy
- F-beta score
- MAPE
- Precision
- Recall
- RMSE
- ROC AUC

Benchmarking Dimensions & Criteria

Figure 7: Benchmarking Criteria relevant for the energy anomaly detection use case
Own representation

To understand whether criteria are relevant to the *Production Inc.* use case, the individual dimensions are discussed in more detail. Since data quality has a positive effect on learning processes and the performance of a ML model, criteria related to the nature of a data basis should generally have a sufficient degree of quality. Therefore, for the use case of *Production Inc.* all criteria related to the dimension *Data* are considered as relevant. Only the criterion *Timeliness* can be neglected for the use case to a limited extent. In general, the prediction of anomalies in energy consumption is not related to a temporal component as represented by the criterion. Moreover, its expression has no significant influence on the occurrence of energy anomalies. However, if data sets are used for algorithm training that go back further into the past and are thus outdated with respect to technical progress, the criterion of *Timeliness* should be considered. For example, data from a time of older equipment generations that had not yet implemented current standards would not be sufficient for training. Therefore, the *Timeliness* of data in this use case is of limited relevance depending on advancement cycles of the underlying equipment.

Since it is a goal for *Production Inc.* to reduce *Energy Consumption* and the associated *Computation Cost* in operational terms, possible algorithms are to be compared regarding their characteristics. Therefore, the selected algorithm should not consume more energy and incur more costs than it can save through its energy anomaly predictions and corresponding measures. In general, the balance of savings and consumption should be kept. However, the criterion of *Computation Time* can be neglected to a limited extent. On the part of the productive application no ad-hoc answers are necessary since the specific energy anomaly prediction is not time-critical. If the predictions are made in a time contingent individual to the user, the application is sufficient. In case of computation times exceeding the time contingent, such as several days and weeks, the energy anomaly prediction proves to be insufficient. *Energy Consumption* would be negatively affected, contrary to the underlying goals. Therefore, the criterion of *Computation Time* is of limited relevance for the use case.

Regarding the *Explainability* of the energy anomaly prediction, the relevance of the criteria for the use case of *Production Inc.* must be considered in a differentiated way. In view of the circumstances brought about by environmental regulations, major internal company transformation processes are necessary, involving large amount of investment. Therefore, special consideration must be given to the prediction performance achieved when selecting algorithms. However, the *Trust* placed in the respective algorithm by the users must also be considered as important, given the impact of the predictions on the company. The criterion of *Accessibility* is to be judged as relevant in the context of the algorithm selection process, as it enables a continuous improvement of the model. In addition, new insights into the energy anomaly predictions can be gained through adjustments made, which in turn increases the confidence in the application. To be able to use the knowledge gained by the algorithms, such as the degree of use of renewable energy sources or the number of emission certificates to be acquired, the criteria of *Informativeness* and *Causality* should be considered. By fulfilling the criterion of *Transferability*, the flexibility of the productive application should also be preserved to be able to make correct predictions in new environments, such as after the procurement of new equipment with differently structured data output. The criterion of *Fairness* can be neglected since the underlying use case leaves no room for discrimination. The *Proxy Functionality* can also be neglected since no further application-specific features are required in terms of the explanatory power of the prediction model.

In terms of the relevance of prediction performance metrics, the underlying task is to classify labeled data according to common properties and thus make predictions about anomalies in energy consumption. Accordingly, only the classification metrics are relevant for the *Production Inc.* use case. The regression metrics *MAPE* and *RMSE* can be neglected.

To be able to determine the value of the artifacts according to Gregor und Hevner, the evaluation is concluded with statements to their *Validity*, *Utility*, *Efficacy*, and *Quality* (2013). As shown by the underlying use case, the artifacts embodied by the benchmarking model and the list of benchmarking-relevant criteria are applicable according to the criterion of *Validity* from a theoretical point of view. Thus, in terms of *Utility*, best solutions can be achieved in a systematic way in productive applications. Regarding to the criteria *Efficacy* and *Quality*, no statement can yet be made from a purely theoretical perspective in view of the still pending real-world application. This evaluation step also helps to make more detailed statements about *Validity* and *Utility* regarding their characteristics.

6. Discussion

Discussing the artifacts with research experts and applying them to the use case presented allows us to address practical implications of the underlying research. First, we ran a structured approach to ML project implementation that provides a step-by-step guideline for practitioners from initial motivations to evaluation of results. As part of the approach, SML algorithms are benchmarked against each other so that users can identify the best algorithm solution for their underlying use case. By weighing up possible solutions, the user can achieve a deeper understanding of the use case and the project task. Second, we provide a holistic overview of dimensions and criteria that are relevant for benchmarking SML algorithms and for the subsequent application. This provides the basis for comparing individual SML algorithms with respect to their performance according to predefined criteria. Since the dimensions and criteria are generic, as is the underlying procedure, their listing represents a toolbox that is used differently for individual use cases. Using the overview of benchmarking-relevant criteria, project managers can, for example, formulate concrete requirements for productive application. Third, the developed artifacts, embodied by the benchmarking model and the listed criteria included in the procedure, represent so-called meta-models, which can be supplemented by individual methods. For example, the Sustainable Machine Learning Balance Sheet can be used in the operation dimension to compare the energy savings achieved by a ML application with the energy consumption from model training as well as productive operation (Wenninger et al. 2022). Thus, the developed artifacts provide a standardized and at the same time flexible approach to the user. The generic character can be applied to ML efforts in multiple areas. In summary, when applied in practice, our models can have a significant positive impact on project results and generate business value, be it in the development of productive solutions or in addressing prevailing challenges in the context of ML.

However, our research as well as the promises in terms of practical improvements are also associated with limitations. Although our approach is to select SML algorithms as objectively as possible without situational tendencies by using predefined dimensions and criteria, a so-called *status quo bias* of the users cannot be completely prevented by our structured approach. Hence, in the *Algorithm Selection* step, prior knowledge about certain algorithms already influences which algorithms are compared in terms of *Benchmarking* and which algorithm possibly performs best. Since the number of algorithms and their characteristics changes continuously, a comprehensive listing of possible algorithms would not be of long prevalence (Caruana and

Niculescu-Mizil 2006). In addition, the *Design & Development* step does not consider whether the underlying model is sufficiently trained. Accordingly, the case of model overfitting can occur with a high number of training cycles, so that the model delivers high prediction performance on the training data but is not generic enough for the application on unknown data. Lastly, for the specific applicability of the overview on benchmarking-relevant criteria, as in the case of energy anomaly detection, expert advice may be necessary to identify the criteria important for the use case.

In terms of further research, several challenges arise that can still be addressed. The first challenge is the individual weighting of benchmarking-relevant criteria in the sense of a quantitative decision model. This would allow the user to weight the existing overview of criteria using a gradation with respect to the importance of individual criteria. The need for individual weighting by the user could thus be eliminated. In addition, in view of the focus on SML the development of benchmarking models for UML and RL algorithms would be pending, whose steps in *Design & Development* and the associated performance metrics differ in comparison to SML. Also, testing the artifacts for validity on federated learning – a learning approach bound on local computers – can still be explored. Another aspect of further research, which will be addressed in the follow-up of this thesis, is the evaluation of underlying artifacts in the real world. Accordingly, different SML algorithms are to be compared with each other under consideration of the benchmarking model when using the same data set. Based on the benchmarking-relevant criteria the best SML algorithm shall be identified. The real-world application should also provide information about the criteria of *Quality* and *Efficacy*, which cannot yet be determined in the context of the theoretical evaluation of this thesis.

7. Conclusion

Since there are many ML algorithms available today and their number is continuously increasing, the question of which algorithm is most suitable for the underlying use case arises. Our research addresses this challenge through the development of two artifacts designed to assist practitioners in solving prevailing problems. The first artifact aims to provide a structured process to make SML algorithms comparable in the sense of *Benchmarking*. The second artifact provides criteria with respect to which the algorithms can be compared objectively. Based on the predefined criteria, the most suitable SML algorithm for an individual use case can be identified. Since we are among the first to create a solution space for the underlying problem, we follow a DSR approach to ensure practical applicability by means of continuous evaluation schemes. Accordingly, the developed artifacts are especially aimed at practitioners and researchers to give them a step-by-step approach to implement their specific ML project and generate business value. In general, we propose a paradigm shift in the selection of algorithms, shifting the focus from the prediction performance of ML algorithms to a holistic consideration of application-relevant factors. With respect to productive solutions, key decisions such as the selection of a specific algorithm cannot be made based on individual dimensions alone. By applying the developed artifacts according to the specifications listed in this thesis, a positive impact on projects in research and practice can be achieved and business value represented by powerful productive solutions can be generated.

References

Agarwal, R. (2019): The 5 Classification Evaluation metrics every Data Scientist must know. In *Towards Data Science*, 9/17/2019. Available online at https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226, checked on 12/30/2021.

Agrawal, A.; Gans, J.; Goldfarb, A. (2018): Prediction Machines. The Simple Economics of Artificial Intelligence: Harvard Business Press.

Agrawal, P.; Arya, R.; Bindal, A.; Bhatia, S.; Gagneja, A.; Godlewski, J. et al. (2019): Data Platform for Machine Learning. In *Proceedings of the 2019 International Conference on Management of Data*, pp. 1803–1816. Available online at https://dl.acm.org/doi/10.1145/3299869.3314050.

Agrawal, S.; Jain, P. (2017): An improved approach for movie recommendation system. In 2017 International Conference on IoT in Social, Mobile, Analytics and Cloud, pp. 336–342.

Askham, N.; Cook, D.; Doyle, M.; Fereday, H.; Gibson, M.; Landbeck, U. et al. (Eds.) (2013): The six primary dimensions for data quality assessment. DAMA UK Working Group. Available online at https://silo.tips/download/the-six-primary-dimensions-for-data-quality-assessment, checked on 1/12/2022.

Barredo Arrieta, A.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A. et al. (2020): Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. In *Information Fusion* 58, pp. 82–115. DOI: 10.1016/j.inffus.2019.12.012.

Batini, C.; Cappiello, C.; Francalanci, C.; Maurino, A. (2009): Methodologies for data quality assessment and improvement. In *ACM Computing Surveys* 41 (3), pp. 1–52. DOI: 10.1145/1541880.1541883.

Berente, N.; Gu, B.; Recker, J.; Santhanam, R. (2021): Special Issue Editor's Comments: Managing Artificial Intelligence. In *Management Information Systems Quarterly* 45 (3), pp. 1433–1450. Available online at https://aisel.aisnet.org/misq/vol45/iss3/16.

Biran, O.; Cotton, Courtenay V. (2017): Explanation and Justification in Machine Learning: A Survey. Available online at https://www.semanticscholar.org/paper/Explanation-and-Justification-in-Machine-Learning-%3A-Biran-Cotton/02e2e79a77d8aabc1af1900ac80ceebac20abde4.

Bishop, C. M. (2006): Pattern Recognition and Machine Learning: Springer.

Breiman, L. (2001): Random Forests. In *Machine Learning* 45 (1), pp. 5–32. DOI: 10.1023/A:1010933404324.

Brodley, C.; Smyth, P. (1995): The Process of Applying Machine Learning Algorithms. Available online at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.4543.

Brynjolfsson, E.; Mitchell, T. (2017): What can machine learning do? Workforce implications. In *Science (New York, N.Y.)* 358 (6370), pp. 1530–1534. DOI: 10.1126/science.aap8062.

Burkart, N.; Huber, M. F. (2021): A Survey on the Explainability of Supervised Machine Learning. In *Journal of Artificial Intelligence Research* 70, pp. 245–317. DOI: 10.1613/jair.1.12228.

Caruana, R.; Niculescu-Mizil, A. (2006): An empirical comparison of supervised learning algorithms. In W. Cohen, A. Moore (Eds.): Proceedings of the 23rd international conference on Machine learning. Pittsburgh, Pennsylvania, 6/25/2006 - 6/29/2006. New York, New York, USA: ACM Press, pp. 161–168.

Cawley, G. C.; Talbot, N. L. C. (2010): On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. In *Journal of Machine Learning Research* (11), pp. 2079–2107. Available online at https://www.jmlr.org/papers/volume11/cawley10a/cawley10a.pdf, checked on 12/23/2021.

Collins, C.; Dennehy, D.; Conboy, K.; Mikalef, P. (2021): Artificial intelligence in information systems research: A systematic literature review and research agenda. In *International Journal of Information Management* 60 (C). DOI: 10.1016/j.ijinfomgt.2021.102383.

Döbel, I.; Leis, M.; Vogelgesang, M. M.; Neustroev, D.; Petzka, H.; Riemer, A. et al. (2018): Maschinelles Lernen. Eine Analyse zu Kompetenzen, Forschung und Anwendung. Fraunhofer. Available online at http://publica.fraunhofer.de/dokumente/N-497408.html.

Domingos, P. (2012): A few useful things to know about machine learning. In *Communications of the ACM* 55 (10), pp. 78–87. DOI: 10.1145/2347736.2347755.

Drew, S. (1997): From knowledge to action: the impact of benchmarking on organizational performance. In *Long Range Planning* 30 (3), pp. 427–441. Available online at https://www.academia.edu/48772800/From_knowledge_to_action_the_impact_of_benchmarking_on_organizational_performance.

Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P. (1996): From Data Mining to Knowledge Discovery in Databases. In *AIMag* 17 (3), p. 37. DOI: 10.1609/aimag.v17i3.1230.

García-Martín, E.; Rodrigues, C. F.; Riley, G.; Grahn, H. (2019): Estimation of energy consumption in machine learning. In *Journal of Parallel and Distributed Computing* 134, pp. 75–88. DOI: 10.1016/j.jpdc.2019.07.007.

Gartner (2020): 5 Trends Drive the Gartner Hype Cycle for Emerging Technologies. Available online at https://www.gartner.com/smarterwithgartner/5-trends-drive-the-gartner-hype-cycle-for-emerging-technologies-2020, updated on 11/11/2021, checked on 11/11/2021.

Goodfellow, I.; Bengio, Y.; Courville, A. (2016): Deep learning. Cambridge, Massachusetts, London, England: The MIT Press (Adaptive computation and machine learning). Available online at https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=6287197.

Gregor, S.; Hevner, A. R. (2013): Positioning and Presenting Design Science Research for Maximum Impact. In *MIS Quarterly* 37 (2), pp. 337–355. DOI: 10.25300/MISQ/2013/37.2.01.

Gudivada, V.; Apon, A.; Ding, J. (2017): Data Quality Considerations for Big Data and Machine Learning: Going Beyond Data Cleaning and Transformations. Available online at https://www.semanticscholar.org/paper/Data-Quality-Considerations-for-Big-Data-and-Going-Gudivada-Apon/625a9e9822603b79f754c4ce044760f7363b5eb6.

Gustavsson, M.; Ljungberg, J. (2021): Algorithms and Their Work: A Performativity Perspective. In *12th Scandinavian Conference on Information Systems*. Available online at https://aisel.aisnet.org/scis2021/7.

Guyon, I.; Elisseeff, A. (2003): An Introduction to Variable and Feature Selection. In *The Journal of Machine Learning Research* 3 (1), pp. 1157–1182. Available online at https://dl.acm.org/doi/pdf/10.5555/944919.944968, checked on 1/9/2022.

Häckel, B.; Karnebogen, P.; Ritter, C. (2021): AI-based industrial full-service offerings: A model for payment structure selection considering predictive power. In *Decision Support Systems* 152, p. 113653. DOI: 10.1016/j.dss.2021.113653.

Hanley, J.; McNeil, B. (1982): The meaning and use of the area under a receiver operating characteristic (ROC) curve. In *Radiology* 143 (1), pp. 29–36. Available online at https://www.semanticscholar.org/paper/The-meaning-and-use-of-the-area-under-a-receiver-Hanley-McNeil/bc37489e7173c75be152b2fcea35191c68847ab2.

Hardt, M.; Price, E.; Srebro, N. (2016): Equality of Opportunity in Supervised Learning. Available online at https://arxiv.org/pdf/1610.02413.

Harvard Business Review (1987): How to Measure Yourself Against the Best. Available online at https://hbr.org/1987/01/how-to-measure-yourself-against-the-best, updated on 8/1/2014, checked on 12/3/2021.

Hevner; A. R.; March, S. T.; Park, J.; Ram, S. (2004): Design Science in Information Systems Research. In *MIS Quarterly* 28 (1), pp. 75–105. Available online at https://www.jstor.org/stable/25148625?seq=1#metadata_info_tab_contents.

Hill, C.; Bellamy, R.; Erickson, T.; Burnett, M. (2016): Trials and tribulations of developers of intelligent systems: A field study. In: 2016 IEEE Symposium on Visual Languages and Human-Centric Computing. Cambridge, 04.09.2016 - 08.09.2016: IEEE, pp. 162–170.

Hoffman, R. R.; Mueller, S. T.; Klein, G.; Litman, J. (2018): Metrics for Explainable AI: Challenges and Prospects. Available online at https://arxiv.org/pdf/1812.04608.

Janiesch, C.; Zschech, P.; Heinrich, K. (2021): Machine learning and deep learning. In *Electronic Markets* 31 (3), pp. 685–695. DOI: 10.1007/s12525-021-00475-2.

Jordan, M. I.; Mitchell, T. M. (2015): Machine learning: Trends, perspectives, and prospects. In *Science (New York, N.Y.)* 349 (6245), pp. 255–260. DOI: 10.1126/science.aaa8415.

Kaymakci, C.; Wenninger, S.; Sauer, A. (2021): Energy Anomaly Detection in Industrial Applications with Long Short-term Memory-based Autoencoders. In *Procedia CIRP* 104, pp. 182–187. DOI: 10.1016/j.procir.2021.11.031.

Kessler, R.; Gómez, J. M. (2020): Implikationen von Machine Learning auf das Datenmanagement in Unternehmen. In *HMD* 57 (1), pp. 89–105. DOI: 10.1365/s40702-020-00585-z.

Ketter, W.; Peters, M.; Collins, J.; Gupta, A. (2016): Competitive Benchmarking: An IS Research Approach to Address Wicked Problems with Big Data and Analytics. In *MISQ* 40 (4), pp. 1057–1080. DOI: 10.25300/MISQ/2016/40.4.12.

Kotsiantis, S. B. (2007): Supervised Machine Learning: A Review of Classification Techniques. In *Informatica* (31), pp. 249–268. Available online at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.95.9683.

Kotsiantis, S. B.; Zaharakis, I. D.; Pintelas, P. E. (2006): Machine learning: a review of classification and combining techniques. In *Artificial Intelligence Review* 26 (3), pp. 159–190. DOI: 10.1007/s10462-007-9052-3.

Kratsch, W.; Manderscheid, J.; Röglinger, M.; Seyfried, J. (2021): Machine Learning in Business Process Monitoring: A Comparison of Deep Learning and Classical Approaches Used for Outcome Prediction. In *Business & Information Systems Engineering* 63 (3), pp. 261–276. DOI: 10.1007/s12599-020-00645-0.

Kühl, N.; Hirt, R.; Baier, L.; Schmitz, B.; Satzger, G. (2021): How to Conduct Rigorous Supervised Machine Learning in Information Systems Research: The Supervised Machine Learning Report Card. In *CAIS* 48 (1), pp. 589–615. DOI: 10.17705/1CAIS.04845.

Lehnhoff, S.; Staudt, P.; Watson, R. (2021): Changing the Climate in Information Systems Research. In *Business & Information Systems Engineering* 63 (3), pp. 219–222. Available online at https://aisel.aisnet.org/bise/vol63/iss3/1.

Li, X. H.; Cao, C. C.; Shi, Y.; Bai, W.; Gao, H.; Qiu, L. et al. (2020): A Survey of Data-driven and Knowledge-aware eXplainable AI. In *IEEE Transactions on Knowledge and Data Engineering* 34 (1). DOI: 10.1109/TKDE.2020.2983930.

March, S. T.; Smith, G. F. (1995): Design and natural science research on information technology. In *Decision Support Systems* 15 (4), pp. 251–266. DOI: 10.1016/0167-9236(94)00041-2.

Microsoft (2021): Was ist der Team Data Science-Prozess (TDSP)? - Azure Architecture Center. Available online at https://docs.microsoft.com/de-de/azure/architecture/data-science-process/overview, updated on 12/1/2021, checked on 12/1/2021.

Mohseni, S.; Zarei, N.; Ragan, E. D. (2021): A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems. In *ACM Trans. Interact. Intell. Syst.* 11 (3-4), pp. 1–45. DOI: 10.1145/3387166.

Müller, O.; Junglas, I.; vom Brocke, J.; Debortoli, S. (2016): Utilizing big data analytics for information systems research: challenges, promises and guidelines. In *European Journal of Information Systems* 25 (4), pp. 289–302. DOI: 10.1057/ejis.2016.2.

Novakovic, J.; Veljovic, A.; Ilić, S.; Papic, Ž.; Milica, T. (2017): Evaluation of Classification Models in Machine Learning. In *Theory and Applications of Mathematics & Computer Science* (7), p. 39. Available online at https://www.uav.ro/applications/se/journal/index.php/TAMCS/article/view/158.

Peffers, K.; Tuunanen, T.; Rothenberger, M. A.; Chatterjee, S. (2007): A Design Science Research Methodology for Information Systems Research. In *Journal of Management Information Systems* 24 (3), pp. 45–77. DOI: 10.2753/MIS0742-1222240302.

Polyzotis, N.; Roy, S.; Whang, S. E.; Zinkevich, M. (2018): Data Lifecycle Challenges in Production Machine Learning. In *SIGMOD Rec.* 47 (2), pp. 17–28. DOI: 10.1145/3299887.3299891.

Rautu, R. S.; Racoviteanu, G.; Dinet, E. (2017): Use of Benchmarking For the Improvement of the Operation of the Drinking Water Supply Systems. In *Procedia Engineering* (209), pp. 180–187. DOI: 10.1016/j.proeng.2017.11.145.

Ribeiro, M. T.; Singh, S.; Guestrin, C. (2016): Why Should I Trust You?": Explaining the Predictions of Any Classifier, pp. 1135–1144. Available online at https://arxiv.org/pdf/1602.04938.

Roelofs, R. (2019): Measuring Generalization and Overfitting in Machine Learning. Electrical Engineering and Comuter Scieneces University of California at Berkeley. Available online at https://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-102.pdf, checked on 1/9/2022.

Rosenfeld, A. (2021): Better Metrics for Evaluating Explainable Artificial In-telligence: Blue Sky Ideas Track. In: AAMAS, 3th - 7th May. Virtual Event, United Kingdom. Available online at https://www.researchgate.net/publication/349111351_Better_Metrics_for_Evaluating_Explainable_Artificial_In-telligence_Blue_Sky_Ideas_Track.

Russell, S. J.; Norvig, P. (2016): Artificial intelligence. A modern approach. With assistance of Ernest Davis, Douglas Edwards. Third edition, Global edition. Boston, Columbus, Indianapolis: Pearson (Always learning). Available online at https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=5831883.

Schaaf, N.; Wiedenroth, S. J.; Wagner, P. (2021): Erklärbare KI in der Praxis - Anwendungsorientierte Evaluation von XAI-Verfahren. Edited by Thomas Bauernhansl, Marco Huber, Werner Kraus. Fraunhofer IPA. Available online at https://publica.fraunhofer.de/eprints/urn_nbn_de_0011-n-6306675.pdf.

Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; Phillips, T.; Ebner, D. et al. (2014): Machine Learning: The High Interest Credit Card of Technical Debt. In *SE4ML*. Available online at https://research.google/pubs/pub43146/.

Sonnenberg, C.; vom Brocke, J. (2012): Evaluation Patterns for Design Science Research Artefacts. In M. Helfert, B. Donnellan (Eds.): Practical Aspects of Design Science, vol. 286. Berlin, Heidelberg: Springer Berlin Heidelberg (Communications in Computer and Information Science), pp. 71–83.

Speicher, T.; Heidari, H.; Grgic-Hlaca, N.; Gummadi, K. P.; Singla, A.; Weller, A.; Zafar, M. B. (2018): A Unified Approach to Quantifying Algorithmic Unfairness: Measuring Individual & Group Unfairness via Inequality Indices. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2239–2248. Available online at https://dl.acm.org/doi/10.1145/3219819.3220046.

Studer, S.; Bui, T. B.; Drescher, C.; Hanuschkin, A.; Winkler, L.; Peters, S.; Müller, K. R. (2021): Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology 3 (2), pp. 392–413. DOI: 10.3390/make3020020.

Teuteberg, F.; Kluth, M.; Smolnik, S.; Ahlemann, F. (2009): Semantic Benchmarking of Process Models - An Ontology-Based Approach. In *ICIS 2009 Proceedings*. Available online at https://aisel.aisnet.org/icis2009/89.

Watson, G. H. (1993): Strategic benchmarking. How to rate your company's performance against the world's best. New York: Wiley.

Wederhake, L.; Wenninger, S.; Wiethe, C.; Fridgen, G.; Stirnweiß, D. (forthcoming): Benchmarking Building Energy Performance: Accuracy by involving occupants in collecting data - A case study in Germany.

Wenninger, S.; Kaymakci, C.; Wiethe, C.; Römmelt, J.; Baur, L.; Häckel, B.; Sauer, A. (2022): How Sustainable is Machine Learning in Energy Applications? - The Sustainable Machine Learning Balance Sheet. In: 17th Conference on Wirtschaftsinformatik. Nürnberg, Germany, February 21th - 23th.

Wenninger, S.; Wiethe, C. (2021): Benchmarking Energy Quantification Methods to Predict Heating Energy Performance of Residential Buildings in Germany. In *Business & Information Systems Engineering* (63), pp. 223–242. Available online at https://www.semanticscholar.org/paper/Benchmarking-Energy-Quantification-Methods-to-of-in-Wenninger-Wiethe/de2c682062dd60ade9b13755be2b19d2042c32d0.

Wirth, R.; Hipp, J. (2000): CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th ICKDDM*, pp. 29–39. Available online at https://www.researchgate.net/publication/239585378_CRISP-DM_Towards_a_standard_process_model_for_data_mining.

Yu, L.; Liu, H. (2004): Efficient Feature Selection via Analysis of Relevance and Redundancy. In *JMLR* (5), pp. 1205–1224. Available online at https://www.jmlr.org/papers/volume5/yu04a/yu04a.pdf, checked on 1/9/2022.

Zaharia, M.; Chen, A.; Davidson, A.; Ghodsi, A.; Hong, S.; Konwinski, A. et al. (2018): Accelerating the Machine Learning Lifecycle with MLflow. In *IEEE*. Available online at https://www.semanticscholar.org/paper/Accelerating-the-Machine-Learning-Lifecycle-with-Zaharia-Chen/b2e0b79e6f180af2e0e559f2b1faba66b2bd578a.

Zhou, J.; Gandomi, A. H.; Chen, F.; Holzinger, A. (2021): Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics. In *Electronics* 10 (5), p. 593. DOI: 10.3390/electronics10050593.

Zuccarelli, E. (2021): Performance Metrics in ML - Part 2: Regression. In *Towards Data Science*, 1/4/2021. Available online at https://towardsdatascience.com/performance-metrics-in-machine-learning-part-2-regression-c60608f3ef6a, checked on 12/30/2021.

Appendix A

Measurement of the *Prediction Performance Metric* ROC-AUC

The *Area Under the Receiver Operating Characteristic Curve (ROC AUC)* is a metric to determine the performance of a classifier. Graphically, the *ROC* curve plots the *True Positive Rate (TPR)* against the *False Positive Rate (FPR)*. By determining the area under the *ROC* curve, the separation ability of a classifier to distinguish between classes at different thresholds can be illustrated (Hanley and McNeil 1982; Novakovic et al. 2017). An exemplary ROC curve is shown in the following figure.

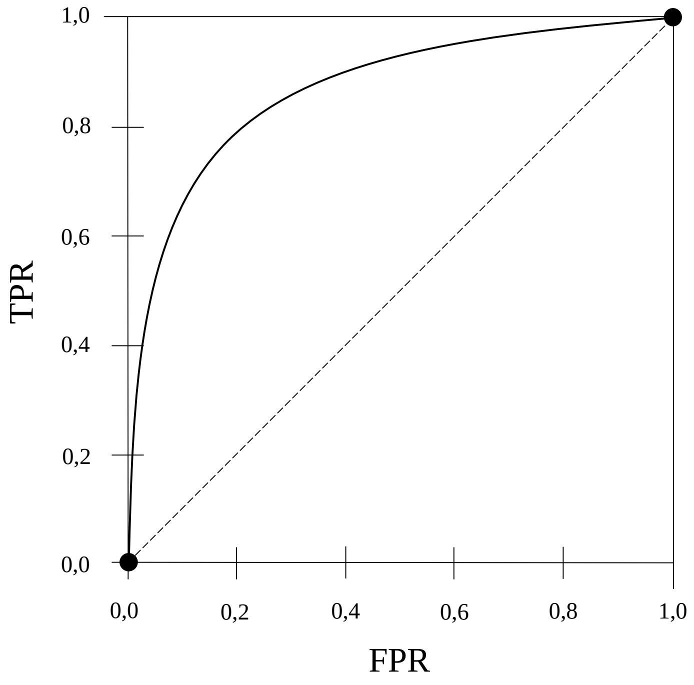

Figure A.1: Exemplary ROC-Curve
(Agarwal 2019)

The corresponding values of *TPR* and *FPR* are calculated as follows:

$$TPR = \frac{TP}{(TP + FN)}$$

$$FPR = \frac{FP}{(TN + FP)}$$

Using the *AUC*, *ROC* curves can be compared with each other. The larger the *AUC*, the better the corresponding algorithm (Novakovic et al. 2017).

Appendix B

Measurement of the *Explainability* Criterion *Trust*

Trust in a ML application results from technical knowledge, following beliefs, and aspects of experience. The criterion includes specific factors such as the reliability, familiarity, and traceability. With the help of a questionnaire, the degree of fulfillment of the factors can be queried accordingly by means of specifically tuned questions (Hoffman et al. 2018). With the aid of scales ranging from strong disagreement (1) to strong agreement (5), values can be assigned to the individual factors, which help to make the criterion of *Trust* measurable.

(1) Does the ML application seem reliable to you regarding the fulfillment of the tasks at hand?

1	2	3	4	5

(2) Are the predictions/results of the ML application comprehensible from your point of view?

1	2	3	4	5

(3) Are you familiar with the decision-making processes underlying the ML application?

1	2	3	4	5

(4) Are actions prevailing in the ML application predictable from your perspective?

1	2	3	4	5

(5) Do you think the ML application consistently shows high performance?

1	2	3	4	5

The points of the individual questions are added together. The higher the total score, the better the *Trust* in relation to a given ML model.

Appendix C

Measurement of the *Explainability* Criterion *Informativeness*

ML Models can help people make decisions through their predictions. In doing so, it is indispensable that people have essential information based on which a ML model acts (Barredo Arrieta et al. 2020). Since the information content of a ML model depends on the algorithm used and the domain knowledge of a human, questions are necessary to help the user determine the *Informativeness* of the underlying ML model (Hoffman et al. 2018; Li et al. 2020). Analogous to the questionnaire for the criterion of *Trust*, the individual questions can again be equipped with scales ranging from strong disagreement (1) to strong agreement (5).

(1) Is detailed information about the chain of reasoning to the final prediction traceable?

1	2	3	4	5

(2) Can new information be gained through the ML model in relation to the use-case at hand?

1	2	3	4	5

(3) Is the information provided by the ML model sufficient to justify decisions to be made?

1	2	3	4	5

The points of the individual questions are added together. The higher the total score, the better the *Informativeness* of a given ML model. In general, transparent models should be rated with a higher *Informativeness*, due to the low effort required for the acquisition of information. However, in the case of black-box models, whose level of information can only be increased by XAI methods such as LIME[1], their nature as an opaque decision model should have a negative impact on the degree of *Informativeness*.

[1] LIME involves modifying a dataset to identify effects on the output of a ML model and to infer its internal working mechanisms (Ribeiro et al. 2016).