Bus Inf Syst Eng
https://doi.org/10.1007/s12599-023-00842-7

RESEARCH PAPER

The Impact of Resource Allocation on the Machine Learning Lifecycle

Bridging the Gap between Software Engineering and Management

Sebastian Duda · Peter Hofmann · Nils Urbach · Fabiane Völter · Amelie Zwickel

Received: 14 May 2022 / Accepted: 11 September 2023

Abstract An organization's ability to develop Machine Learning (ML) applications depends on its available resource base. Without awareness and understanding of all relevant resources as well as their impact on the ML lifecycle, we risk inefficient allocations as well as missing monopolization tendencies. To counteract these risks, the study develops a framework that interweaves the relevant resources with the procedural and technical dependencies within the ML lifecycle. To rigorously develop and evaluate this framework the paper follows the Design Science Research paradigm and builds on a literature review and an interview study. In doing so, it bridges the gap between the software engineering and management perspective to advance the ML management discourse. The results extend the literature by introducing not yet discussed but relevant resources, describing six direct and indirect effects of resources on the ML lifecycle, and revealing the resources' contextual properties. Furthermore, the framework is useful in practice to support organizational decision-making and contextualize monopolization tendencies.

Keywords ML management · Machine learning lifecycle · Artificial intelligence · Resource-based view · Design science research

1 Introduction

The current momentum in Machine Learning (ML) development and adoption is making companies to reflect on their positioning and the associated configuration of their resource bases. Some companies try to stand out with leading ML models (e.g., OpenAI with ChatGPT) or to capture the market with resource-integrating service platform offerings (Geske et al. 2021). Others heavily invest in data collection as data is a critical resource when training an ML model (Mikalef and Gupta 2021). We can even

Accepted after two revisions by Stefan Stieglitz.

S. Duda · P. Hofmann · N. Urbach · F. Völter (✉)
Fraunhofer Institute for Applied Information Technology FIT
Branch Business & Information Systems Engineering,
Wittelsbacherring 10, 95444 Bayreuth, Germany
e-mail: fabiane.voelter@fit.fraunhofer.de

S. Duda
e-mail: sebastian.duda@fit.fraunhofer.de

P. Hofmann
e-mail: p.hofmann@appliedai.de

N. Urbach
e-mail: nils.urbach@fim-rc.de

S. Duda · P. Hofmann · F. Völter
FIM Research Center, University of Bayreuth, Wittelsbacher
Ring 10, 95444 Bayreuth, Germany

P. Hofmann
appliedAI Initiative GmbH, Freddie-Mercury-Straße 5,
80797 Munich, Germany

N. Urbach
Frankfurt University of Applied Sciences, Nibelungenplatz 1,
60318 Frankfurt am Main, Germany

A. Zwickel
University of Bayreuth, Universitätsstraße 30, 95445 Bayreuth,
Germany
e-mail: amelie-zwickel@hotmail.de

Published online: 14 November 2023

observe that infrastructure resources are no longer necessarily a commodity, but companies are scrambling to develop and deploy specific hardware, such as tensor processing units (Jouppi et al. 2018).

As with other digital technologies, companies face once again the challenge of finding their place in the market and, correspondingly, configuring their resource base. Above all, ML applications are most notable in that their performance evolution is non-deterministic because ML "is a subfield of AI, which tries to acquire knowledge by extracting patterns from raw data and solve some problems using this knowledge" (Giray 2021, p. 2). Recent research has acknowledged ML development's specificity (Giray 2021; Iansiti and Lakhani 2020; Kumeno 2020; de Souza Nascimento et al. 2020) and begun investigating which resources are relevant to achieving business value through artificial intelligence (AI) in general (Mikalef and Gupta 2021) or more technology-specifically through ML (Ashmore et al. 2021; Amershi et al. 2019; Idowu et al. 2021).

However, we lack explanations for how resource allocations affect the ML lifecycle, specifically in light of the resources' interwovenness manifesting in both horizontal and vertical ways Horizontally, resources must integrate into the specific ML lifecycle activities. The ML lifecycle structures all activities required to develop, train, and deploy ML models (Hummer et al. 2019). Therefore, companies that arbitrarily invest in resources or that give everyone a slice of the cake do not necessarily solve the specific process problems that they face in the ML lifecycle. Vertically, resources must functionally integrate into the technology stack. For instance, training an ML model requires the appropriate data and infrastructure.

This knowledge gap of not understanding how the provisioning or usage of resources affect the ML lifecycle causes two concerns: For one, the knowledge gap limits companies' ability to make sound strategic decisions considering the configuration of their resource bases and the resulting potential for ML use and offerings. In specific, a recent survey identified that a lack of internal resources represents one of the most common pitfalls for successful ML adoption. However, at the same time, many organizations are lacking the knowledge of how to overcome the challenge of configuring their internal resources for successful ML adoption (Hartmann et al. 2019). IBM (2021) found that 7 out of 10 organizations cannot account for returns on their investments. Also, we might overlook monopolization tendencies or resource dependencies, limiting fair access to transformative technology. In consequence, practical evidence demonstrates that tech companies such as Microsoft or Google are at the ML forefront, while in-house development is difficult or even impossible for others (Davenport 2018). In sum, we lack relevant knowledge for efficient and effective resource allocation decisions throughout the ML lifecycle although they are of strategic relevance for ML adoption. Thus, we ask:

> How do resource allocations impact the ML lifecycle?

To answer our research question, we followed the design science research (DSR) paradigm (Hevner et al. 2004; Peffers et al. 2007) and designed and evaluated a framework that introduces and conceptualizes the relevant resources and their effects on the ML lifecycle. Specifically, we applied the DSR process of Peffers et al. (2007), executing five design iterations. Within the design process, we relied on knowledge gathered from a literature review and 12 expert interviews. Regarding our theoretical lens, we followed the resource based-view (RBV) (Barney 1991; Grant 1991; Powell 1992) as it is suitable for assessing the strategic value of resources (Bharadwaj 2000; Melville et al. 2004).

Our research bridges the gap between the software engineering discourse (Amershi et al. 2019) and the AI/ML management discourse (Berente et al. 2021; Buxmann et al. 2021). The software engineering discourse offers in-depth insights into the processes and technical dependencies within the ML lifecycle that could benefit the AI/ML management discourse. Thus, we contribute to the ML management discourse by providing a software engineering-informed framework that systematizes ML resources as well as their effects on the ML lifecycle. The users of the framework could be both organizational decision-makers and politicians who may use the framework to understand monopolization tendencies and foster the democratization of technology access.

The rest of the paper is structured as follows. In the following section, we present the foundations of our work, and subsequently outline the research method used in this paper. Thereafter, we present the resulting framework including the resources and their effects before presenting the evaluation of the results. Afterward, we discuss our results and conclude in the last section.

2 Foundations

Previous knowledge underlies, explains and informs the design of artefacts (Jones and Gregor 2007). In the following, we outline the ML lifecycle as well as the RBV, which serve as justificatory knowledge for the design of our artefact.

2.1 Machine Learning Development and Application

Before introducing the foundations of the ML lifecycle (i.e., a horizontal perspective) and the stack model (i.e., a vertical perspective), we present our understanding of ML applications and their development: An ML application "is said to learn from experience E with respect to some class of task T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" (Mitchell 1997, p. 2). Accordingly, ML applications are non-deterministic, but are based on statistical patterns extracted from the data. Enhancements along the ML lifecycle can improve the performance of ML applications (Giray 2021).

From a horizontal perspective, the literature provides different breakdowns of the ML lifecycle's process phases, which differ in their depth and concerning the problem under consideration. The representations of the ML lifecycle have in common that they describe a highly iterative process in which feedback loops (validation and verification) are essential to meet the predefined objectives (Gharibi et al. 2021). Thus, developing ML applications could be "viewed as searching through a large space of candidate programs, guided by training experience, to find a program that optimizes the performance metric" (Jordan and Mitchell 2015, p. 255). While common ML lifecycle process representations differ only in a few details, we draw on Amershi et al.'s (2019) well-acknowledged work. We illustrate our understanding of the ML lifecycle in Fig. 1, as per Amershi et al. (2019). The feedback arrows indicate typical feedback loops. While the arrows pointing to the left describe the loop back to any preceding activity, the upward-pointing arrow describes only the loop between model training and feature engineering (Amershi et al. 2019; Ashmore et al. 2021).

Data collection aims at accumulating and integrating heterogeneous (real-world) data (Baier and Seebacher 2019). The data subsequently undergo *data cleaning* to enhance data quality. For instance, tools support *data cleaning* by clearing wrong or noisy data points (Amershi et al. 2019). In the case of supervised ML, one proceeds with complementing each record with ground truth labels (*data labeling*) (Amershi et al. 2019; Kotsiantis et al. 2006). To prepare the pre-processed data for training the ML model, one proceeds with *feature engineering,* i.e., extracting and selecting informative features (Amershi et al. 2019). Features are a set of attributes, often represented by vectors (Akkiraju et al. 2018; Mohri et al. 2012). However, not all ML models require the same features. For instance, while support vector machines require well-developed features, other models, such as deep learning models, automate this step during ML model training (Amershi et al. 2019; Lins et al. 2021). The *model training* step includes selecting, configuring, and optimizing an ML model (Ashmore et al. 2021; Akkiraju et al. 2018). It is possible to create and train the ML model from scratch or rely on transfer learning to make existing pre-trained ML models applicable in the new domain (Gharibi et al. 2021). In the *model evaluation* step, the ML model's performance with previously defined metrics is evaluated (Amershi et al. 2019). The trained and validated model is then transferred to the target infrastructure in the *model deployment* step (Amershi et al. 2019; Ashmore et al. 2021; Gharibi et al. 2021). In this step, the model is integrated into traditional software and offered to the users (Ashmore et al. 2021). Subsequently, in the model monitoring step, the deployed model is continuously monitored to detect errors during the real-world execution (Gharibi et al. 2021; Amershi et al. 2019).

Besides, our paper relies on the work of Lins et al. (2021) to conceptualize the vertical perspective. While previous authors have investigated stack models in the context of cloud services (Liu et al. 2011; Mell and Grance 2011), we chose Lins et al.'s (2021) model as the authors represent the full depth of functional integration and cover ML services, which makes the model highly suitable for our work. Based on Liu et al.'s (2011) as well as Mell and Grance's (2011) work, Lins et al. (2021) hierarchically distinguish between three layers (software services, developer services, and infrastructure services) and specify their components. While the software services layer includes "ready-to-use AI applications and building blocks", the developer services layer includes all tools "for assisting developers in implementing code" (Lins et al. 2021, p. 442). The infrastructure services comprise computational power as well as network and storage capacities (Lins et al. 2021).

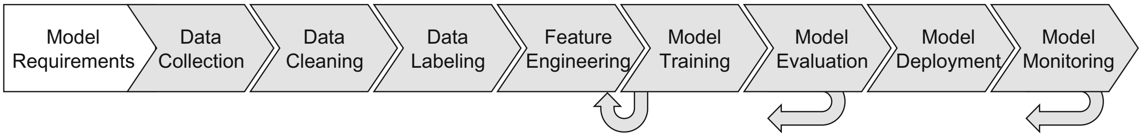

Fig. 1 The machine learning lifecycle (Amershi et al. 2019)

2.2 A Resource-Based View on Machine Learning Resources

The RBV substantiates the origin of sustainable competitive advantage, whereas the object of investigation focuses on the heterogeneity of the resource base of companies (Barney 1991; Grant 1991; Powell 1992; Das and Teng 2000). Accordingly, the RBV is a suitable perspective for assessing the strategic value of resources (Bharadwaj 2000; Melville et al. 2004). Particularly, our research benefits from the conceptualization of resources and the explanation of competitive advantage arising from the ownership or control of resources (Barney 2001). Penrose (1959) and Wernerfelt (1984) define resources as a company's collection of tangible resources such as infrastructure as well as intangible assets such as licenses. However, as Bharadwaj (2000) notes, tangible and intangible resources alone do not create a competitive advantage: Human resources determine an organization's ability to coordinate and integrate tangible and intangible resources. As such, the combination of three resource classes leads to organizational capabilities, which in turn, can result in a competitive advantage: tangible (e.g., physical resources), intangible (e.g., strategy or licenses), and human skills (e.g., technical and managerial skills of employees) (Bharadwaj 2000).

To generate a sustainable competitive advantage, resources must be valuable, rare, immobile, difficult to imitate, and non-substitutable (Peteraf 1993; Barney 1991; Bharadwaj 2000; Powell 1992). If developing the necessary resources in-house is too costly or time-consuming, companies may draw on resources controlled by others (Madhok 1997). For instance, emerging AI service platforms aim to address the need to develop appropriate resources (Geske et al. 2021). However, the RBV does not specify the underlying mechanisms that explain how companies gain a competitive advantage through resources (Melville et al. 2004). As a theoretical answer, research came up with the capabilities concept. Capabilities describe a company's abilities to integrate or combine resources to achieve a competitive advantage (Grant 2010). Capabilities develop through the interaction of resources (Amit and Schoemaker 1993).

In the context of ML, organizations can benefit from resource-oriented decision-making as well. To become and remain competitive, decision-makers need to purposefully allocate available resources to enhance the ML lifecycle, enhance its output, and/or reduce associated costs. Decision-makers need to understand how resources impact the ML lifecycle to consider the consequences of resource allocations. Thus, the effects of resource allocations on the ML lifecycle are relevant for sound strategic decision-making.

However, while the following research has already taken on deriving resources relevant for levering the potential of ML or associated technologies, their impact on the ML lifecycle remains neglected. The existing literature offers only selective insights into the consequences of resource allocations (Shams 2018). In specific, the academic discourse on relevant resources and capabilities for (big) data analytics is remarkably prevalent (Gupta and George 2016; Mikalef et al. 2018). Nonetheless, we cannot only rely on (big) data analytics frameworks since software engineering research suggests a re-investigation, as existing research specifically addresses certain domains and states that "on this topic, many research questions remain unanswered" (Giray 2021, p. 28; de Souza Nascimento et al. 2020; Shimagaki et al. 2018; Kumeno 2020; Wan et al. 2020). Having already explored (big) data analytics resources and capabilities in recent studies, Mikalef and Gupta (2021) have also developed a capabilities model for AI. As the authors "identify several key types of resources" (Mikalef and Gupta 2021, p. 2), they provide a comprehensive overview of resource categories, e.g., technology or data, required for building organizational capabilities. In parallel, Weber et al. (2022) refer to data, AI-specific infrastructure, and IT infrastructure when it comes to technical resources. At the same time, the level of detail in previous research does not allow to draw conclusions with regards to the categories' specific items as well as their dependencies. Also, Papagiannidis et al. (2021) orchestrate AI resources and refer to the resources data, infrastructure, and human skills, as well as further intangibles. As the authors focus on the transformation of resources into capabilities, they do not systematize the dependencies of resources. Thus, we conclude that existing research has successfully identified relevant resources in related fields (e.g., big data analytics), however, we lack explanations on how and where ML-related resources contribute to a company's capabilities to approach the ML lifecycle. Specifically, the current literature does not provide a systematic analysis of the resources needed for ML and their impact on the ML lifecycle. By focusing on the ML lifecycle, we spotlight the ML lifecycle's resource peculiarities including their dependencies and, thus, bridge the gap between established research strands in management (resource-based view) and computer science (software engineering).

3 Research Method

We followed the DSR paradigm, as the approach allows us to rigorously develop and evaluate artifacts (Hevner et al. 2004; Peffers et al. 2007). As such, the DSR paradigm provides us with a methodological frame that enables us to iteratively incorporate both the extant literature and the

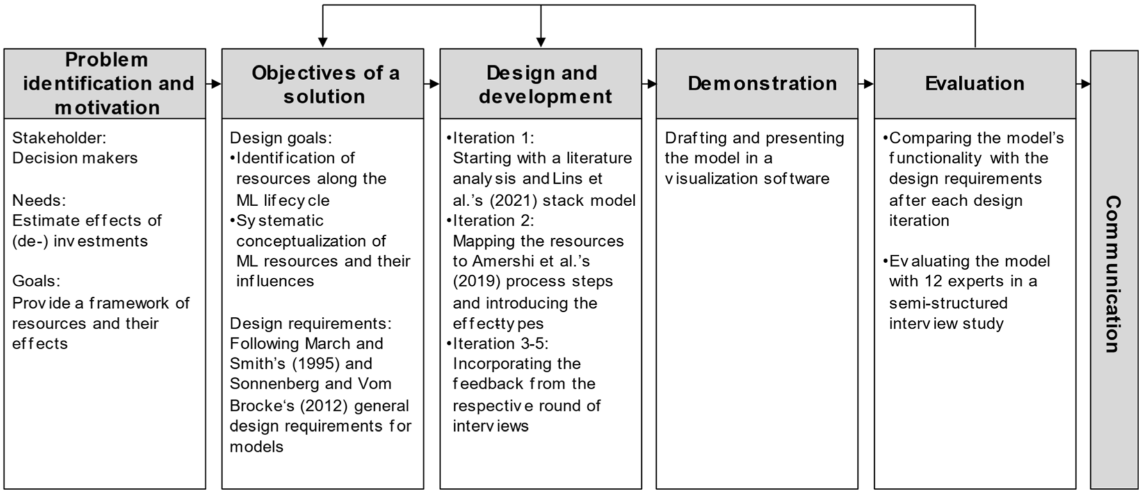
Problem identification and motivation
Stakeholder: Decision makers
Needs: Estimate effects of (de-) investments
Goals: Provide a framework of resources and their effects
Objectives of a solution
Design goals:
•Identification of resources along the ML lifecycle
•Systematic conceptualization of ML resources and their influences
Design requirements: Following March and Smith's (1995) and Sonnenberg and Vom Brocke's (2012) general design requirements for models
Design and development
•Iteration 1: Starting with a literature analysis and Lins et al.'s (2021) stack model
•Iteration 2: Mapping the resources to Amershi et al.'s (2019) process steps and introducing the effecttypes
•Iteration 3-5: Incorporating the feedback from the respective round of interviews
Demonstration
Drafting and presenting the model in a visualization software
Evaluation
•Comparing the model's functionality with the design requirements after each design iteration
•Evaluating the model with 12 experts in a semi-structured interview study
Communication

Fig. 2 Overview of the research method

expert knowledge of decision-makers in the rigorous design and evaluation of an artifact whose relevance stems from solving real-world problems (Hevner 2007). We applied the DSR process of Peffers et al. (2007), executing five design iterations (see Fig. 2 as well as Appendix A. Appendices available online via http://link.springer.com).

The DSR process of Peffers et al. (2007) starts with problem identification and motivation followed by the definition of the solution objectives. While we refer to the paper's introduction for the detailed problem description and motivation, we summarize the problem space according to Maedche et al. (2019) as follows:

- **Stakeholders**: We aim to support decision-makers who make resource allocation decisions regarding the ML lifecycle as well as politicians who want to foster the democratization of technology access.
- **Needs**: Organizational and political decision-makers need the ability to make sound decisions considering the configuration of organizational resource bases. However, in order to do so, decision-makers need to know what resources are potentially relevant and how they affect the ML lifecycle.
- **Goals**: We aim to identify and conceptualize the relevant resources and their impact on the ML lifecycle in a consistent and useful framework.
- **Requirements**: Our framework should meet the evaluation criteria for models (the level of detail, internal consistency, fidelity with real-world phenomena, robustness, completeness, and understandability) as per March and Smith (1995) as well as Sonnenberg and vom Brocke (2012).

In the following, we first describe how we conducted our data collection and analysis. We then describe the design iterations in detail.

3.1 Data Collection and Analysis

3.1.1 Literature Review

We started our methodological process from scratch and collected justificatory knowledge from a literature review to inform the artifact design. For identifying relevant literature, we followed the guidelines of Webster and Watson (2002) and carried out a structured literature search. The search string consists of two parts: The first part ensures the focus on ML and associated terms. The second part ensures the focus on resources and service offerings related to ML. We used the following search string:

> ("machine learning" OR "artificial intelligence" OR "deep learning")
> AND
> ("resource based view" OR "resource-based view" OR "value network" OR "resource orchestration" OR "resource dependency" OR "business value" OR "infrastructure as a service" OR "infrastructure-as-a-service" OR "inference-as-a-service" OR "inference as a service" OR "machine learning as a service" OR IaaS OR "machine learning as a service" OR AIaaS OR "software engineering").

We searched in the databases Web of Science (WoS), the Association for Information Systems eLibrary (AISeL), Business Source Premier, IEEE Xplore, and the Association for Computing Machinery (ACM). The selected databases hold both technical and business research papers, which is necessary to identify the entire variety of resources. We applied the inclusion criteria of (1) English papers published in (2) journals or conference proceedings. The resulting literature search initially yielded a total of 3,024 papers with duplicates (WoS: 1,521; AISeL: 33;

IEEE: 705; Business Source Premier: 413; ACM: 352). Also, papers should (3) not represent a duplicate. The latter inclusion criteria resulted in a set of 2,354 papers. During the subsequent screening of the title, abstract, and full-text, we applied the exclusion criteria of relevance. We deemed papers relevant if they included a differentiated consideration of resources needed throughout the ML lifecycle. During the title and abstract screening, we removed 2,278 papers, leaving a number of 76 papers. After applying our exclusion criterion during the full-text screening, we excluded 38 further papers. By forward and backward search, we identified six additional papers. Accordingly, our final set includes 44 papers. The concept matrix used for analyzing the final data set is shown in Appendix A.

3.1.2 Expert Interviews

To access new knowledge and to further evaluate the framework, we conducted an interview study. Thus, practitioners' feedback through a qualitative interview study (expert evaluations) represents the basis for the artifact revisions in Iterations 3, 4, and 5. The interview guide consists of four parts: (1) We started with an opening and introduction, in which both parties introduced themselves, the interviewer clarified organizational conditions (e.g., audio recording, anonymity), and briefly introducedthe research project's goal. (2) The interviewer then asked which resources the interviewee utilized in past ML projects. (3) In the third part, the interviewer introduced the framework, requested feedback on the interviewee's first impression, and subsequently asked whether the framework fulfills the evaluation criteria. (4) Subsequently, the interviewer asked the interviewee to assess the availability and accessibility of the resources. To ensure the consistency and understandability of our questions as well as a smooth procedure and an appropriate time frame, we pretested the interview guide by conducting a mock interview with a data analytics professional. We used purposive expert sampling in the selection of interviewees (Bhattacherjee 2012).

When beginning to examine the experts' feedback, we took every comment into account even though they had different implications for the framework. The first interview part (opening and introduction) did not yield implications for the framework. When analyzing the second interview part (open question about which resources the interviewee utilized in past ML projects), we first checked whether each resource mentioned was already part of our framework. If that was not the case, we considered the resource(s) in the next design iteration (cf. the description of design iterations and Appendix B). In the third interview part, we asked for feedback on the interviewee's first impression and subsequently assessed whether the framework fulfilled the evaluation criteria. Here, every critical comment wasn taken into account in the next design iteration (cf. the description of design iterations and Appendix B). The fourth interview part (assessing the availability and accessibility of the resources) helped us to better understand the experts' context but did not help us to improve the framework.

3.2 Artifact Design Iterations

3.2.1 Iteration 1: Initial Framework Draft

In the following full-text analysis, we extracted all resources and influencing factors mentioned in the papers. Therefore, we started with a broad categorization of resources according to data, infrastructure/hardware, technical implementation (i.e., resources needed for the execution of the learning process), and a residual category. We used this outline of possible resources to familiarize ourselves with the subject area for obtaining an overview of the literature. However, we revised this categorization during each design iteration. In the first design iteration, we decided to adopt the structure of Lins et al. (2021) stack model (see the foundations section). Thus, the first design iteration is based on the hypothesis that the individual resources may be assigned to the service layers of the stack model and are available to the market as a service if they cannot be provided internally. Accordingly, we assigned resources along the categories AI software services, AI developer services, and AI infrastructure services, which in turn consist of AI compute and AI data. In addition, we distinguished whether resources represent an action and or a factor influencing other resources. Therefore, our first draft represented an overview of the resources mentioned in the literature without timely dependencies. We evaluated our first draft with the help of a formative criteria-based evaluation (Cater-Steel et al. 2019; Venable et al. 2016) and applied the evaluation criteria for models proposed by Sonnenberg and vom Brocke (2012). Our evaluation shows that the framework lacked internal consistency (e.g., resources on different abstraction levels or an inconsistent conceptualization of effects) and comprehensibility (e.g., unclarity of how to read the framework or information overload), as well as a coherent level of detail. Moreover, the first draft did not fulfill the evaluation criteria of completeness (e.g., the framework did not cover all activities of the ML lifecycle and resources' effects).

3.2.2 Iteration 2: Mapping of Resources Along the ML Lifecycle

Based on these drawbacks, we further overhauled the framework during the second iteration. Most importantly, we improved the inconsistency of effects between resources, e.g., by introducing two effect classes (direct and indirect effects) and six effect types (supplementing, iterating, reusing, automating, informing, creating). Furthermore, we horizontally mapped resources according to the nine process steps of an ML lifecycle as proposed by Amershi et al. (2019). In parallel to the first iteration, we conducted a formative evaluation method by applying a criteria-based evaluation (Cater-Steel et al. 2019; Venable et al. 2016). The latter showed that while the introduction of classes and types as well as the structuring along the ML lifecycle allowed us to improve the information overload, the framework still faced some drawbacks regarding understandability. Furthermore, we could not yet comprehensively map the resources along the ML lifecycle. Thus, the second iteration of the framework still faced several drawbacks, especially understandability, comprehensibility, internal consistency, and varying level of detail.

3.2.3 Iteration 3–5: Incorporating the Feedback from the Interviews

We conducted an interview study to access new knowledge and to further evaluate the framework. In particular, we interviewed six experts during Iteration 3, two experts during Iteration 4, and four experts during Iteration 5 (see Table 1). Thus, experts evaluated an updated framework after Iterations 3 and 4.

Based on the feedback of E1–E6, we revised the demarcation of resources (e.g., introducing the glue/reusable code and division of tools into more precise functional groups), rearranged data resources, and specified the types of relationships. We evaluated the third iteration's results with E7 and E8. During the fourth iteration, we reworked the vertical arrangement of the resources, simplified the framework's presentation, specified resources in primary and secondary resources, and introduced missing resources (e.g., data repositories and data generation tools). During the fifth iteration, the feedback of E9-E12 required slight adjustments of the existing tools and the addition of human skills and strategy, resulting in the framework as described in the results section. The changes conducted in each iteration are summarized in Appendix B.

4 Results

4.1 Overview of the Framework

Before we describe the framework's resources and effects in more detail, we first provide an overview of the framework (see Fig. 3). The framework's purpose is to introduce relevant resources and describe their effects throughout the ML lifecycle. Hence, we provide descriptive and explanatory knowledge for the configuration of an organization's resource base to leverage the ML lifecycle. Although the framework is considered a blueprint for the configuration of an organization's resource base, we emphasize that organizations should apply the framework specifically to their context.

The framework arranges primary and secondary resources horizontally according to the ML lifecycle (see Sect. 2). Accompanying the ML lifecycle, we acknowledge people- and business-related resources. **Primary resources** (grey boxes) are involved as input and output factors in the ML lifecycle (i.e., the improvement of one resource positively affects the succeeding resource). The primary resources match the ML lifecycle starting with the

Table 1 Overview of the Experts

Phase	Expert	Position	Years of experience	Duration
Iteration 3	E1	Senior ML Software Engineer	6 years	52 min
	E2	ML Engineer	4 years	54 min
	E3	Senior Product Manager for ML	6 years	53 min
	E4	ML Team Lead & Product Manager	3 years	52 min
	E5	Data Analyst	5 years	48 min
	E6	CEO, ML startup	6 years	50 min
Iteration 4	E7	Team Lead, Data Science	5 years	56 min
	E8	Team Lead, Data Science	5 years	44 min
Iteration 5	E9	Data Scientist	3 years	33 min
	E10	Senior Manager, Digital Transformation Data & Analytics	9 years	37 min
	E11	ML Engineer	2 years	41 min
	E12	Team Lead, Data Science	8 years	56 min

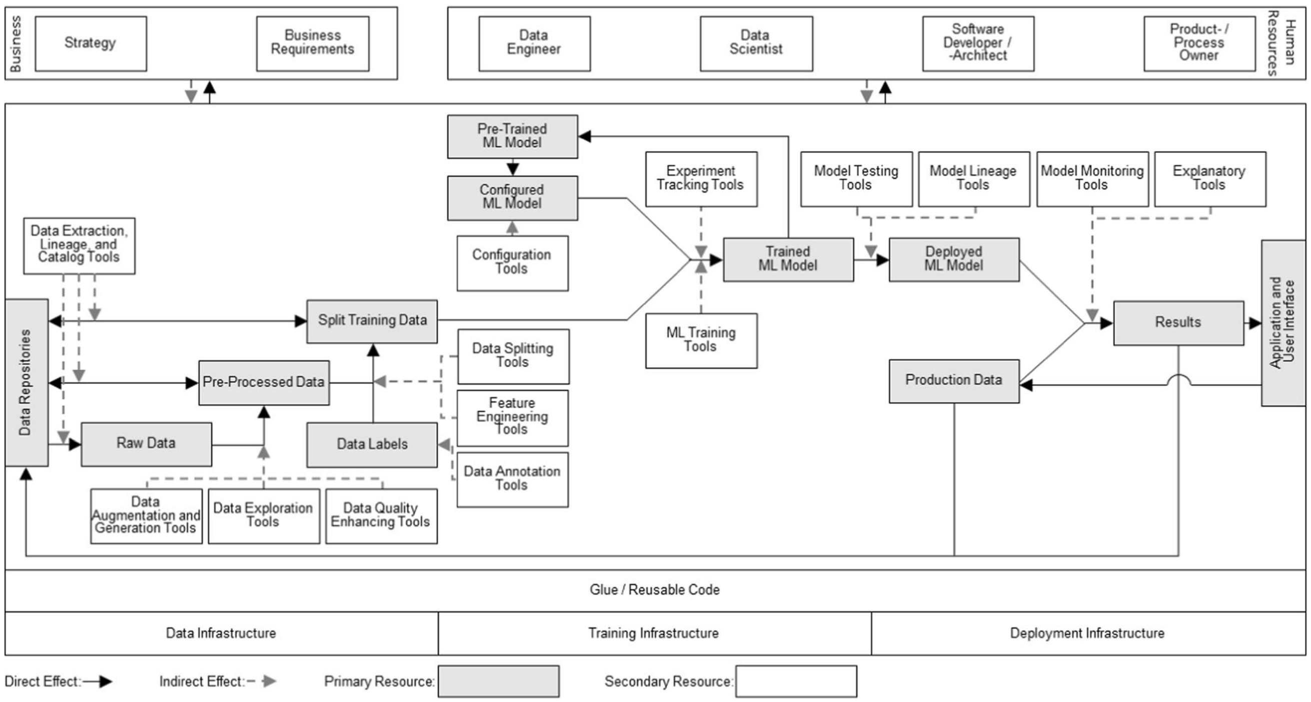

Fig. 3 The machine learning effects framework

collection of raw data through to provisioning the results to the user. The ML lifecycle does not have a singular direction, but may iterate at various points, depicted with the arrows in the left direction. **Secondary resources** (white boxes) affect the ML lifecycle's ability to transform input to output resources.

We follow Bharadwaj's (2000) conceptualization of relevant resources for the creation of competitive advantages, thus, our framework considers tangible resources (e.g., hardware resources), intangible resources (e.g., the organization's strategy), and human resources (e.g., data scientists). Following Bharadwaj (2000), human resources are specifically relevant for coordinating and integrating tangible and intangible ML resources. Thus, despite varying in volume, the graphical size is not related to the importance of human resources.

Furthermore, the value-creating dependencies along the lifecycle reflect a temporal dependency. While a clear estimation of the required time to move along the lifecycle is highly context-dependent, research highlights that most time is dedicated to data management (Haakman et al. 2021; Abubakar et al. 2020).

For mapping the effects of resources (illustrated as arrows), we introduce different effect classes: **Direct effect classes** (solid arrows) connect the primary resources along the ML lifecycle. The arrow's direction indicates its value-creating direction. Thus, one might move back and forth along the process relationships, as the ML lifecycle's experimental nature expects.

Joints of process relationships symbolize the imperative that two input resources must match (e.g., training data must match the configured or pre-trained ML model). **Indirect effect classes** (dashed arrows) moderate the ML lifecycle's ability to transform input to output resources.

4.2 Resources

In Table 2, we introduce the identified resources and describe them according to their function in the ML lifecycle. Organizations may instantiate a resource differently: For one thing, there might be different approaches to fulfill a purpose (e.g., self-developed code vs. a third-party application). For another thing, an organization might bundle resources differently (e.g., independently implemented data pipelines vs. a full-featured ML service platform). For example, SciKit-Learn[1] provides, among others, the functionality of the configuration tools and the ML training tools.

Following the RBV, the resources' contribution to the ownership or control of a unique and effective resource base for leveraging the ML lifecycle explains the resource's strategic value. We do not only consider the existence of resources but their contextual properties in the explanation of competitive advantages. When assessing ML

[1] SciKit-Learn (scikit-learn.org) is an ML library in Python that among others supports various ML algorithms and feature engineering tools and also provides raw data.

Table 2 Definitions of the identified resources

Resource	Explanation
Data Repositories	Internal and external storage of (un-)structured data
Raw Data	Input data in a form as originally retrieved from its source
Pre-Processed Data	Cleaned, transformed, and normalized data with features suitable for training a configured ML model
Data Labels	Informative tags describing individual data points for supervised learning
Split Training Data	Ready-to-use data for the training of configured ML model (usually) split into three parts: training data, test data, and validation data
Pre-Trained ML Models	A configured ML model that has been trained and can be further trained with additional data
Configured ML Model	The training algorithm and a set of related hyperparameters that specify the structure and learning process of an ML model
Trained ML Model	A configured ML model with parameter values learned from data
Deployed ML Model	A ML model that can be used for predictions in the production environment
Production Data	The data suitable for the deployed ML model for prediction
Results	The outcome of the prediction based on the production data
Application and User Interface	The interface to an application or to the user that allows one to display the deployed ML model's output and/or input production data
Data Extraction, Lineage, and Catalog Tools	Tools that allow one to retrieve, maintain data, and trace their processing journey and its influencing factors
Data Augmentation and Generation Tools	Tools to modify existing and to create new data for training a configured ML model
Data Exploration Tools	Tools to investigate the existing data to improve data understanding (e.g., visualization)
Data Quality Enhancing Tools	Tools that detect and handle missing, noisy and invalid data to enhance their quality for training a configured ML model
Data Annotation Tools	Tools that allow one to annotate and process raw data into a form suitable for training a configured ML model
Feature Engineering Tools	Tools that allow one to extract, select, and construct features suitable for training a configured ML model
Data Splitting Tools	Tools that provide approaches and algorithms to split the data into training, test, and validation data
Configuration Tools	Tools helping to select well-performing hyperparameters for ML models to enhance their performance
ML Training Tools	Tools that allow one to perform the learning process suitable for the configured ML model
Experiment Tracking Tools	Tools that allow one to trace all information arising from the experimental process of training an ML model
Model Testing Tools	Tools that test the functionality and performance of models
Model Lineage Tools	Tools that allow one to trace trained ML models' development journey and its influencing factors
Model Monitoring Tools	Tools that allow one to track and assess the deployed ML models' activity
Explanatory Tools	Tools that provide additional information that explains the result to the user
Glue / Reusable Code	Supporting and reusable code, which manages tools and resources along the ML lifecycle
Data Infrastructure	An infrastructure fulfilling the storage and computation requirements of the ML lifecycle's activities and tools for managing data
Training Infrastructure	An infrastructure fulfilling the storage and computation requirements of the ML lifecycle's activities and tools for training and verifying the model
Deployment Infrastructure	An infrastructure fulfilling the storage and computation requirements of the ML lifecycle's activities and tools for deploying and running the ML application

resources, we emphasize their context-dependency for three reasons:

First, the contribution of resources to leveraging the ML lifecycle depends on the context. For example, a business intelligence (BI) department might not need model monitoring tools when they only perform a one-time ad-hoc analysis.

Second, we also recommend refraining from making context-independent statements about resource properties beyond the value property (i.e., rareness, imitation difficulty, or non-substitutability) as our interviewees' assessment of resources' characteristics varied highly. For instance, while E8 highlighted that appropriate data is rare, E11 negates data rareness.

Third, the capabilities literature (Amit and Schoemaker 1993; Grant 2010) underpins the necessity to consider the interaction of resources and, thus, their integration or combination. We will present the identified effect classes and effects resulting from resource interactions in detail in the next section.

4.3 Effects in the Framework

We identified two superordinate effect classes, **direct** and **indirect,** systematizing the influences between resources throughout the ML lifecycle. Direct effects describe the influence of resources on other resources. Indirect effects describe the moderation of relationships between other resources. Before explaining each effect, we provide a summary of the effects in Table 3. The column 'Examples' lists the resources that exert the respective effect.

4.3.1 Direct Effects

Resources may directly affect succeeding resources in the ML lifecycle in three ways: supplementing, iterating, and reusability effects.

4.3.1.1 Supplementing Supplementing effects throughout the ML lifecycle have been well-researched in the past. For example, Amershi et al. (2019) and Arpteg et al. (2018) point out that resources directly influence the succeeding resources' value. For example, split training data serves as input for the training of ML models and, thus, directly affects the subsequent resource. As another example, the infrastructure directly affects the ML lifecycle throughout its phases. For instance, while managing the data, the data infrastructure provides both storage capacity and computational power for data processing (Jöhnk et al. 2021; Mikalef et al. 2019).

> "What helps us are data pipelines: The data infrastructure hosts all structured data we have in databases. [..] Also, we have data lakes where we can just dump the data." (E8).

4.3.1.2 Iterating Iterating effects constitute in resources affecting other resources in succeeding iterations of the same ML lifecycle. As a result, iterating effects improve a product/outcome developed in an ML lifecycle. Unlike all other effects, the iterating effects' value-creating direction plays out in succeeding **iterations** in the ML lifecycle. For example, production data can be stored in the internal data repository after its use. As a result, the stored production data increases the amount of available raw data for the re-training of the ML model in a subsequent iteration of the ML lifecycle. Thus, ML models can be continuously improved over the course of an ML lifecycle.

> "What we also see often is productive data, which at the same time is also fed back into pre-processed data to serve as new features [..] That's why it can also flow directly into pre-processed data." (E8).

4.3.1.3 Reusability Reusability effects differ from iterative effects in terms of which ML lifecycle they influence: While iterating effects play out in iterations of the same ML lifecycle, **reusability** effects influence another ML lifecycle. For example, pre-trained models from previous ML lifecycles can be used and improve the development of other products (Yang et al. 2017). As a result, in contrast to

Table 3 Overview of the Identified Forms of Effects

Effect class	Effects	Direction	Examples (non-exhaustive)
Direct effect	Supplementing	Resources serving as input for succeeding resources within an ML lifecycle iteration	All gray resources along the ML lifecycle; infrastructure
	Iterating	Resources affecting resources in the succeeding iterations of the same ML lifecycles	Production data; results
	Reusability	Resources affecting resources in other ML lifecycles	Production data; trained ML models
Indirect effect	Automating	Resources automating process steps	Feature engineering tools; data catalog, extraction, and lineage tools, data quality-enhancing tools; configuration tools, ML training tools; model testing tools
	Extending	Resources extending or creating new resources	Data augmentation and generation tools; data annotation tools
	Informing	Resources providing additional information	Data exploration tools; experiment tracking tools; model lineage tools; model monitoring tools, explanatory tools

iterating effects, reusability effects do not continuously improve a specific product but reinforce the value of available resources for developing further ML applications as these can rely on resources from previous ML lifecycles. Our interviewees highlighted the fact that an integrated development of resources is beneficial as storing and reusing resources leads to significant time and cost savings:

> "If you can first take a pre-trained model and adapt it, your own development time is simply shorter until you see whether a use case works." (E7).

4.3.2 Indirect Effects

Besides directly affecting other resources, resources may also indirectly influence the ML lifecycle by moderating the relationship between resources. For example, improving feature engineering tools enhances the ability to generate *Split Training Data* based on pre-processed data. As resources indirectly influence the ML lifecycle, they support organizations in their ability to use and integrate resources and, thus, enhance their organizational capabilities for ML development (Kogut and Zander 1992; Peppard and Ward 2004). We distinguish between three indirect effects: automating, informing, and extending.

4.3.2.1 Automating First, tools may indirectly influence the ML lifecycle by **supporting the automation of tasks**. For example, *data quality-enhancing tools* support developers in cleaning and preparing data for the model learning and verification (Haakman et al. 2021; Kumeno 2020), e.g., by conducting automatic fixes such as automated suggestions for missing values (Reimann and Kniesel-Wünsche 2020). Similarly, *Data Extraction, Catalog, and Lineage Tools* support managing data dependencies, e.g., by running automated checks to ensure that dependencies are annotated and by visualizing dependency trees (Sculley et al. 2015):

> "To somehow ensure that the data arrives in this data lake in very good quality and that it is regularly kept up to date, that it is accessible, usable; this is incredibly valuable if it's the case." (E12).

4.3.2.2 Informing Furthermore, tools may increase the attention developers pay to errors that cannot be fixed automatically (Polyzotis et al. 2018), which represents the second categorization of indirect effects: **providing** additional **information**. On the one hand, tools providing additional information support understanding the status quo (e.g., *Data Exploration Tools, Model Monitoring Tools, Explanatory Tools*) as E10 highlights:

> "I first integrate all the data [..] and analyze it visually to get a first grasp of the data." (E10).

On the other hand, tools providing additional information support iterations or subsequent ML lifecycles by tracking and managing changes over the course of ML lifecycle (e.g., *Experiment Tracking Tools, Model Lineage Tools*). As such, tracking and lineage tools can help developers to make the exploitation of iterating and reusability effects more effective. For example, *Data Extraction, Catalog, and Lineage Tools* support the management of continuously added data:

> "This is a very, very, very, very important tool, [so] that I know when I have done all this data management and pre-processing and something: How are different data actually connected? What types of transformations have happened? So that I can transparently track the whole [extract-transform-load] routes. This is very important." (E10).

4.3.2.3 Extending Third, indirect effects **extend** or **create new resources**. Thus, resources with an indirect extending effect represent the basis for the creation of other resources (e.g., ML models generalize from training data). For example, *Data Annotation Tools* help to create the resource *data labels* (Agrawal et al. 2019; Amershi et al. 2019; Haakman et al. 2021; Whang and Lee 2020). Similarly, *Data Augmentation & Generation Tools* allow for generating additional data. As a result, they support developers in improving the accuracy of ML models (Polyzotis et al. 2018; Whang and Lee 2020):

> "[..] you actually also do data generation to specifically generate edge cases, so we see whether our model can handle this, what the limit is, when it breaks." (E8).

5 Evaluation

We evaluate the proposed framework as proposed by Hevner et al. (2004) and Peffers et al. (2007). We further follow the evaluation criteria for DSR artifacts as proposed by Sonnenberg and vom Brocke (2012). Accordingly, we validated the framework's completeness, understandability, level of detail, robustness, internal consistency, and fidelity with real-world phenomena (March and Smith 1995; Sonnenberg and vom Brocke 2012).

Overall, the experts mostly stated that the framework is **understandable**. To achieve this, we improved improved minor aspects accordingly throughout the iterations. Several experts, however, agree that explanations would be helpful for understanding (E1, E2, E7, E8). During iteration five, all interview partners agreed on the unreserved understandability of the framework. In the third iteration, some interviewees found that the framework required a higher **level of detail** (E1, E3, E6). Thus, throughout the iteration, we particularized some resources to align the level of detail across the framework. As a result, during iterations four and five, five out of six interview partners stated that the level of detail was appropriate. The interviewees also agreed on the fulfillment of the criteria **robustness**, as the experts stated that the framework is universally applicable across domains and applications. Also, we extended the framework to capture various perspectives on ML, such as BI units, which use ML models' results for internal business processes rather than exposing them to customers. In parallel, the experts agreed that the framework is **internally consistent**. Regarding the framework's **completeness**, the interviewees highlighted some additional resources used in practice, which we had not identified in the reviewed literature, i.e., data lineage tools, data generation tools (E8), and data exploration tools (E9).

As providing utility represents the utmost goal of design science researchers (Hevner et al. 2004), we also considered the **fidelity with real-world phenomena** in the evaluation**.** Specifically, we evaluated the problem statement as well as the framework's applicability. The interviewed experts also agreed with the problem statement. Accordingly, a lack of knowledge regarding the relationships and interconnectedness of the resources throughout the ML lifecycle prevails. As E7 puts it, there is "nothing to hold on to" when it comes to identifying how one's organization is positioned. As a consequence, ML often is reduced to the model's training and verifying. In contrast, managing data is often ignored in hiring and resource allocation decisions even though it provides the foundation for the subsequent ML lifecycle (E9). Against this backdrop, most interviewees agreed that the framework enables the consideration and evaluation of individual resources. For example, E4 noted that the framework can serve as a tool for evaluating the required resources to set up an ML service from a business owner's perspective. With regard to the application of the framework, the interviewed experts agreed that the framework would support them in understanding their organization's positioning against the backdrop of the dependencies of resources in the ML lifecycle (E7, 8, 9, 10, 11). In particular, they stated that they could apply the framework to analyze their organization's portfolio to then subsequently discuss the next steps and resource allocations (E7), especially during the planning of an upcoming project (E10). For example, E8 explains that the framework represents a shopping list for organizations. Thus, we conclude that our framework provides utility for solving real-world problems. However, E9 points out that the organization's maturity regarding ML affects its ability to apply the framework. While the interviewees pointed out the framework's capacity to generate a portfolio view, they also highlighted the complexity of organizational decision-making regarding strategic resource allocations. Accordingly, the framework cannot capture the complexity in terms of the requirements of the underlying organizations (E1, E3, E5) as well as regarding the dimensions of a certain resource (E4, E5).

6 Discussion

Making organizational or political decisions about resources relevant to ML requires a solid understanding of resource requirements and their impact on the ML lifecycle. Without awareness of all relevant resources as well as their effects, we risk inefficient resource allocations as well as missing monopolization tendencies. To address this concern, we have broken new ground by building on software engineering knowledge from the literature and experts to extend the AI/ML discourse (Berente et al. 2021; Buxmann et al. 2021). We incorporate resource requirements into the ML lifecycle by interweaving the identified resources with the procedural and technical dependencies within the ML lifecycle. As a result, we provide a novel perspective on resource allocations that is both relevant for strategic decision-making and applicable to the processes actually taking place in organizations. Existing ML lifecycle models, such as Ashmore et al. (2021), mainly focus on the activities taking place during the development and deployment of an ML application. In contrast, our framework describes the resources and their effects on the ML lifecycle. We argue that our framework does not only provide utility for practitioners by solving practice-oriented problems, but also provides utility to other researchers by theorizing on resources and their effects. In specific, we advance the extant literature in three ways:

First, our identified resources extend the list of relevant resources previously discussed in the context of (big) data analytics (Gupta and George 2016; Mikalef et al. 2018) or AI (Mikalef and Gupta 2021; Weber et al. 2022). The newly found resources resulted either from ML-specifics (e.g., pre-trained model or ML training tools) or an applied software engineering perspective. Notably, our framework is rich and specific in the secondary resource of tools, which predominantly resulted from our interviewees' insights and have been neglected so far in extant literature. As such, our framework emphasizes that one should take tools into account that enable humans to create and coordinate resources and not only consider hiring more experts.

Second, we contribute to the theory body by introducing and describing direct and indirect effects classes, which can each take the form of three effects along the ML lifecycle. Theorizing on resources' effects allows us to understand the consequences of resource allocation, bundling, and scaling as we categorize and explain the implications and long-term consequences caused by specific resources. For example, feeding back production data in data repositories may allow for increasing the accuracy of ML models in subsequent ML lifecycles. However, this effect only comes into play when an organization scales its ML output. In contrast, if an organization is unable to manage its trained ML models, these models cannot be fed into a stock of pre-trained ML models in subsequent ML lifecycles, limiting the available resources for training. This example highlights that the marginal utility of resources is not necessarily linear. As such, organizational decision-making regarding the resource allocation should never be free from internal contextualities, such as the existing organizational resource portfolio, as well as from external contextualities, such as self-reinforcing effects.

Third, our observation through the RBV lens revealed that there is no standard resource base for leveraging the ML lifecycle to gain competitive advantages. Instead, we underpinned the importance of considering a resource's contextual properties. Specifically, the application domain and purpose affect the value, rareness, inimitability, and substitutability of resources.

In order to allow researchers and practitioners to quickly grasp the above-mentioned implications, we summarize them as follows:

1. A context-specific resource portfolio perspective is important for purposeful resource allocation decisions, as there are technical and procedural dependencies between resources along the ML lifecycle. However, resource allocation decisions should not only consider the isolated effects of resources, but also their integration via glue code.
2. The technical and procedural dependencies determine six effect types, which differ regarding the affected resource(s) or the activities in the ML lifecycle and the time at which value is created.
3. The scope of resource allocation decisions must not be limited to the current ML lifecycle iteration but may also affect succeeding ML lifecycle iterations and other ML lifecycles.
4. Investments in secondary resources do not necessarily add value to primary resources due to the non-deterministic nature of ML model development, but increase the effectiveness and efficiency of experimentation and, thus, the likelihood of value creation.

From a managerial perspective, our framework reduces the uncertainty in organizational and political decision-making. We reduce the risk of inefficient resource allocations that may result from being unaware of relevant resources and the specific effects of a resource on the ML lifecycle (i.e., the resource's specific potential to either solve or cause a software engineering problem) as well as the risk of ignoring the context-dependency of the identified resources. Our framework serves as a decision-supporting model of the ML lifecycle's resource space but is not a blueprint for resource allocation decisions. In contrast, it serves to benchmark organizations' positioning regarding the availability and accessibility of resources. Specifically, organizations can "consider which components there are, how the organizations [are] positioned, and where [they] currently stand" (E7). Hence, our framework guides the assessment of an organization's readiness and maturity for developing and deploying ML applications.

Moreover, our systematic categorization offers indications regarding the democratization of ML resources. While our interviewees did not indicate access restrictions at this point in time, our findings make it possible to deduce the strategic importance of specific resources and the implications if their access were restricted in the future. For example, an organization that can rely on a large stack of trained ML models from previous iterations can generate more accurate results in the future. Thus, based on our findings, we predict that continuous development of ML applications scales in terms of a widening gap in available resources between large- and small-scale customers. Besides these demand-sided advantages of hyper-scalers, we also found indications for supply-sided advantages. For example, interviewee E8 indicated that infrastructure service providers prioritize demands of large-scale customers for additional infrastructure capacities over small-scale customers. This prioritization affects the flexibility of small-scalers (e.g., startups) to cope with increasing demand on their side. Additionally, E7 worries about the market power of the hyper-scalers that provide the infrastructure in the cloud. "You can quickly become a pawn of the big players. They can raise prices from one day to the next, comparable to gasoline prices" (E7). Thus, we conclude that the democratization of ML resources does not represent a major challenge just yet but will become of increasing importance with the further development of ML applications and an according increase in ML resources and capabilities.

7 Conclusion

We followed a DSR approach involving a systematic literature review and 12 expert interviews to design and evaluate a framework that introduces and conceptualizes the relevant resources and their effects on the ML lifecycle. We discovered different classes of direct and indirect effects. The direct effect class can take the form of supplementing, iterating, and reusability effects. Indirect effect classes may affect relationships between resources in three different ways: by providing additional information, automating process steps, or extending resources.

Our study contributes to the academic discourse in IS research on the management of the ML lifecycle. While Mikalef et al. (2018) and Weber et al. (2022) explored capabilities for AI, the current academic discourse on the ML lifecycle lacks a portfolio view of ML resources, which includes the interaction and integration of resources. This lack of knowledge causes uncertainty in organizational as well as political decision-making when it comes to efficient resource allocation decisions and the detection of monopolization tendencies. To bridge this gap, we provide theoretical insights into how the effects of ML resources can be systematized. In specific, we explain how resource allocation decisions affect the ML lifecycle. This theoretical knowledge guides decision-making regarding the sourcing, bundling, and scaling of resources.

Despite following a rigorous research approach, our study is subject to limitations, which are mostly reflected in our method of data collection. First, we rely on a literature review as well as interviews for the identification of ML resources. While this allows us to provide a portfolio view of the resources as well as their effects, it limits our findings to the function and features of resources. Future studies could conduct a deep dive, e.g., by conducting case studies, into organizations using ML, which would make it possible to observe the use of resources in action and, thus, derive the form and configuration of resources. Second, we rely on a purposive interview sample of 12 experts who work for organizations developing ML applications. While we identified the importance of context-specific consideration of resources, our sampling did not allow us to derive further insights concerning the factors explaining this context dependency. An explicit focus on sampling specific types of organizations would enable future researchers to address this limitation: For example, laggers which are unable or have failed to develop in-house ML applications, would further enrich our findings. In parallel, an explicit focus on organizations with limited means, such as startups, would provide further insights into the criticality of certain resources. Specifically, we expect that an explicit focus on these organizations would further sharpen our understanding of resources representing a barrier to the market entry of ML development and, thus, the democratization of the access to ML resources.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s12599-023-00842-7.

Acknowledgements We gratefully acknowledge the Bavarian Ministry of Economic Affairs, Regional Development and Energy for their support of the project "Fraunhofer Blockchain Center (20-3066-2-6-14)" that made this research possible.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Aboueata N, Alrasbi S, Erbad A, Kassler A, Bhamare D (2019) Supervised machine learning techniques for efficient network intrusion detection. In: Proceedings of the 28th international conference on computer communication and networks. IEEE, New York

Abubakar H, Obaidat MS, Gupta A, Bhattacharya P, Tanwar S (2020) Interplay of machine learning and software engineering for quality estimations. In: Proceedings of the 2020 international conference on communications, computing, cybersecurity, and informatics, Sharjah, pp 1–6

Agrawal P, Arya R, Bindal A, Bhatia S, Gagneja A, Godlewski J, Low Y, Muss T, Paliwal MM, Raman S, Shah V, Shen B, Sugden L, Zhao K, Wu M-C (2019) Data platform for machine learning. In: Proceedings of the 2019 international conference on management of data. ACM, New York, pp 1803–1816

Amershi S, Begel A, Bird C, DeLine R, Gall H, Kamar E, Nagappan N, Nushi B, Zimmermann T (2019) Software engineering for machine learning: a case study. In: Proceedings of the 41st international conference on software engineering: software engineering in practice, Montreal, pp 291–300

Amit R, Schoemaker PJH (1993) Strategic assets and organizational rent. Strateg Manag J 14:33–46. https://doi.org/10.1002/smj.4250140105

Arpteg A, Brinne B, Crnkovic-Friis L, Bosch J (2018) Software engineering challenges of deep learning. In: Proceedings of the 44th Euromicro conference on software engineering and advances applications. IEEE, New York, pp 50–59

Ashmore R, Calinescu R, Paterson C (2021) Assuring the machine learning lifecycle: desiderata, methods, and challenges. ACM Comput Surv 54:1–39

Baier L, Seebacher S (2019) Challenges in the deployment and operation of machine learning in practice. In: Proceedings of the 27th European conference on information systems, Stockholm

Balayn A, Lofi C, Houben G-J (2021) Managing bias and unfairness in data for decision support: a survey of machine learning and data engineering approaches to identify and mitigate bias and unfairness within data management and analytics systems. Int J Very Large Data Bases 30:739–768

Barney J (1991) Firm resources and sustained competitive advantage. J Manag 17:99–120

Barney J (2001) Resource-based theories of competitive advantage: a ten-year retrospective on the resource-based view. J Manag 27:643–650. https://doi.org/10.1016/S0149-2063(01)00115-5

Belani H, Vukovic M, Car Z (2019) Requirements engineering challenges in building AI-based complex systems. In: Proceedings of the 27th international requirements engineering conference workshops. IEEE, pp 252–255

Berente N, Gu B, Recker J, Santhanam R (2021) Managing artificial intelligence. MIS Q 45:1433–1450

Bharadwaj AS (2000) A resource-based perspective on information technology capability and firm performance: an empirical investigation. MIS Q 24(1):169–196

Bhattacharjee B, Boag S, Doshi C, Dube P, Herta B, Ishakian V, Jayaram KR, Khalaf R, Krishna A, Li YB, Muthusamy V, Puri R, Ren Y, Rosenberg F, Seelam SR, Wang Y, Zhang JM, Zhang L (2017) IBM deep learning service. IBM J Res Dev 61:1–10

Bhattacherjee A (2012) Social science research: Principles, methods, and practices. Open Textbooks, University of Florida. http://scholarcommons.usf.edu/oa_textbooks/3. Accessed 23 Sep 2023

Buxmann P, Hess T, Bennet Thacher J (2021) AI-based information systems. Bus Inf Syst Eng 63:1–4. https://doi.org/10.1007/s12599-020-00675-8

Cater-Steel A, Toleman M, Rajaeian MM (2019) Design science research in doctoral projects: an analysis of Australian theses. JAIS. https://doi.org/10.17705/1jais.00587

Das T, Teng B-S (2000) A resource-based theory of strategic alliances. J Manag 26:31–61. https://doi.org/10.1016/S0149-2063(99)00037-9

Davenport TH (2018) From analytics to artificial intelligence. J Bus Anal 1:73–80. https://doi.org/10.1080/2573234X.2018.1543535

de Souza Nascimento E, Ahmed I, Oliveira E, Palheta MP, Steinmacher I, Conte T (2019) Understanding development process of machine learning systems: challenges and solutions. In: Proceedings of the 2019 ACM/IEEE international symposium on empirical software engineering and measurement

de Souza Nascimento E, Nguyen-Duc A, Sundbø I, Conte T (2020) Software engineering for artificial intelligence and machine learning software: a systematic literature review. https://arxiv.org/ftp/arxiv/papers/2011/2011.03751.pdf. Accessed 2 Feb 2023

Duong TNB, Sang NQ (2018) Distributed machine learning on IAAS clouds. In: Proceedings of the 5th IEEE international conference on cloud computing and intelligence systems. IEEE, pp 58–62

Fujii G, Hamada K, Ishikawa F, Masuda S, Matsuya M, Myojin T, Nishi Y, Ogawa H, Toku T, Tokumoto S, Tsuchiya K, Ujita Y (2020) Guidelines for quality assurance of machine learning-based artificial intelligence. Int J Softw Eng Knowl Eng. https://doi.org/10.1142/s0218194020400227

Geske F, Hofmann P, Lämmermann L, Schlatt V, Urbach N (2021) Gateways to artificial intelligence: developing a taxonomy for AI service platforms. In: Proceedings of the 29th European Conference on Information Systems (ECIS)

Gharibi G, Walunj V, Nekadi R, Marri R, Lee Y (2021) Automated end-to-end management of the modeling lifecycle in deep learning. Empir Softw Eng. https://doi.org/10.1007/s10664-020-09894-9

Giray G (2021) A software engineering perspective on engineering machine learning systems: State of the art and challenges. J Syst Softw 180:111031. https://doi.org/10.1016/j.jss.2021.111031

Grant R (1991) The resource-based theory of competitive advantage: implications for strategy formulation. Calif Manag Rev 33:114–135. https://doi.org/10.2307/41166664

Grant R (2010) Contemporary strategy analysis: Text and cases, 7th edn. Wiley, Hoboken

Gupta M, George JF (2016) Toward the development of a big data analytics capability. Inf Manag 53:1049–1064. https://doi.org/10.1016/j.im.2016.07.004

Haakman M, Cruz L, Huijgens H, van Deursen A (2021) AI lifecycle models need to be revised: An exploratory study in Fintech. Empir Softw Eng. https://doi.org/10.1007/s10664-021-09993-1

Hazelwood K, Bird S, Brooks D, Chintala S, Diril U, Dzhulgakov D, Fawzy M, Jia B, Jia Y, Kalro A, Law J, Lee K, Lu J, Noordhuis P, Smelyanskiy M, Xiong L, Wang X (2018) Applied machine learning at Facebook: A datacenter infrastructure perspective. In: Proceedings of the IEEE International Symposium on High Performance Computer Architecture. IEEE, pp 620–629

Hesenius M, Schwenzfeier N, Meyer O, Koop W, Gruhn V (2019) Towards a software engineering process for developing data-driven applications. In: Proceedings of the 7th international workshop on realizing artificial intelligence synergies in software engineering. IEEE, pp 35–41

Hevner AR (2007) A three cycle view of design science research. Scand J Inf Syst 19:4–10

Hevner AR, March ST, Park J, Ram S (2004) Design science in information systems research. MIS Q 28(1):75–105

Hill C, Bellamy R, Erickson T, Burnett M (2016) Trials and tribulations of developers of intelligent systems: a field study. In: Blackwell A et al (eds) Proceedings of the IEEE Symposium on visual languages and human-centric computing. IEEE, New York, pp 162–170

Hummer W, Muthusamy V, Rausch T, Dube P, El Maghraoui K, Murthi A, Oum P (2019) ModelOps: cloud-based lifecycle management for reliable and trusted AI. In: IEEE International Conference on Cloud Engineering. Conference Publishing Services, IEEE Computer Society, Tokyo, pp 113–120

Hutchinson B, Smart A, Hanna A, Denton E, Greer C, Kjartansson O, Barnes P, Mitchell M (2021) Towards accountability for machine learning datasets: practices from software engineering and infrastructure. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. ACM, New York, pp 560–575

Iansiti M, Lakhani KR (2020) Competing in the age of AI: How machine intelligence changes the rules of business Competitive strategy. Harv Bus Rev 98(1):60–67

Idowu S, Struber D, Berger T (2021) Asset management in machine learning: a survey. In: Proceedings of the 43rd international conference on software engineering. IEEE Computer Society, Los Vaqueros, pp 51–60

Javadi SA, Cloete R, Cobbe J, Lee MSA, Singh J (2020) Monitoring misuse for accountable 'artificial intelligence as a service'. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. ACM, New York

John MM, Olsson HH, Bosch J (2020) AI on the edge: architectural alternatives. 46th Euromicro Conference on Software Engineering and Advanced Applications. IEEE Computer Society, Los Alamitos, pp 21–28

Jöhnk J, Weißert M, Wyrtki K (2021) Ready or not, AI comes – An interview study of organizational AI readiness factors. Bus Inf Syst Eng 63(1):5–20

Jones D, Gregor S (2007) The anatomy of a design theory. JAIS 8:312–335. https://doi.org/10.17705/1jais.00129

Jordan MI, Mitchell T (2015) Machine learning: trends, perspectives, and prospects. Sci 349:255–260. https://doi.org/10.1126/science.aaa8415

Jouppi N, Young C, Patil N, Patterson D (2018) Motivation for and evaluation of the first tensor processing unit. IEEE Micro 38:10–19. https://doi.org/10.1109/MM.2018.032271057

Kogut B, Zander U (1992) Knowledge of the firm, combinative capabilities, and the replication of technology. Organ Sci 3:383–397. https://doi.org/10.1287/orsc.3.3.383

Kotsiantis S, Kanellopoulos D, Pintelas P (2006) Data preprocessing for supervised learning. Int J Comput Sci 1(1):111–117

Kumeno F (2020) Sofware engineering challenges for machine learning applications: a literature review. Intell Decis Technol 13:463–476. https://doi.org/10.3233/IDT-190160

Lins S, Pandl KD, Teigeler H, Thiebes S, Bayer C, Sunyaev A (2021) Artificial intelligence as a service. Bus Inf Syst Eng 63:441–456. https://doi.org/10.1007/s12599-021-00708-w

Lwakatare LE, Raj A, Crnkovic I, Bosch J, Olsson HH (2020) Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions. Inf Softw Technol. https://doi.org/10.1016/j.infsof.2020.106368

Madhok A (1997) Cost, value and foreign market entry mode: the transaction and the firm. Strateg Manag J 18:39–61

Maedche A, Gregor S, Morana S, Feine J (2019) Conceptualization of the problem space in design science research. In: Tulu B, Djamasbi S, Leroy G (eds) Extending the boundaries of design science theory and practice. Springer, Cham, pp 18–31

March ST, Smith GF (1995) Design and natural science research on information technology. Decis Support Syst 15:251–266. https://doi.org/10.1016/0167-9236(94)00041-2

Mell PM, Grance T (2011) The NIST definition of cloud computing. National Inst Standards Technol. https://doi.org/10.6028/NIST.SP.800-145

Melville N, Kraemer K, Gurbaxani V (2004) Review: information technology and organizational performance: an integrative model of IT business value. MIS Q 28:283. https://doi.org/10.2307/25148636

Mikalef P, Gupta M (2021) Artificial intelligence capability: Conceptualization, measurement calibration, and empirical study on its impact on organizational creativity and firm performance. Inf Manag 58:103434. https://doi.org/10.1016/j.im.2021.103434

Mikalef P, Pappas IO, Krogstie J, Giannakos M (2018) Big data analytics capabilities: a systematic literature review and research agenda. Inf Syst e-Bus Manag 16:547–578. https://doi.org/10.1007/s10257-017-0362-y

Mikalef P, Fjortoft SO, Torvatn HY (2019) Developing an artificial intelligence capability: a theoretical framework for business value. In: Abramowicz W, Corchuelo R (eds) Business Information Systems Workshops, vol 373. Springer, Cham, pp 409–416

Mitchell TM (1997) Machine learning. McGraw-Hill, New York

Mohri M, Rostamizadeh A, Talwalkar A (2012) Foundations of machine learning. MIT Press, Cambridge, Adaptive computation and machine learning

Nguyen G, Dlugolinsky S, Bobák M, Tran V, López García Á, Heredia I, Malík P, Hluchý L (2019) Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. Artif Intell Rev 52:77–124. https://doi.org/10.1007/s10462-018-09679-z

Papagiannidis E, Merete Enholm I, Mikalef P, Krogstie J (2021) Structuring AI resources to build an AI capability: a conceptual framework. In: Proceedings of the 29th European Conference of Information Systems

Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S (2007) A design science research methodology for information systems research. J Manag Inf Syst 24:44–77. https://doi.org/10.2753/MIS0742-1222240302

Penrose E (1959) A resource based view of the firm. Strateg Manag J 5:171–180

Peppard J, Ward J (2004) Beyond strategic information systems: towards an IS capability. J Strateg Inf Syst 13:167–194. https://doi.org/10.1016/j.jsis.2004.02.002

Peteraf MA (1993) The cornerstones of competitive advantage: a resource-based view. Strateg Manag J 14:179–191. https://doi.org/10.1002/smj.4250140303

Philipp R, Mladenow A, Strauss C, Völz A (2020) Machine learning as a service. In: Indrawan-Santiago M et al (eds) Proceedings of the 22nd International Conference on Information Integration and Web-based Applications & Services. ACM, New York, pp 396–406

Polyzotis N, Roy S, Whang SE, Zinkevich M (2018) Data lifecycle challenges in production machine learning. ACM SIGMOD Rec 47:17–28. https://doi.org/10.1145/3299887.3299891

Powell TC (1992) Strategic planning as competitive advantage. Strateg Manag J 13:551–558. https://doi.org/10.1002/smj.4250130707

Reimann L, Kniesel-Wünsche G (2020) Achieving guidance in applied machine learning through software engineering techniques. Conference Companion of the 4th International Conference on Art, Science, and Engineering of Programming. ACM, New York, pp 7–12

Ribeiro M, Grolinger K, Capretz MA (2015) MLaaS: machine learning as a service. In: Proceedings of the 14th international conference on machine learning and applications. IEEE, pp 896–902

Saldamli G, Nishit D, Vishal G, Jainish P, Mihir P, Ertaul L (2021) Analysis of machine learning as a service. In: Proceedings of the 17th international conference on grid, cloud, & cluster computing

Schmidt R, Zimmermann A, Moehring M, Keller B (2020) Value creation in connectionist artificial intelligence – a research agenda. In: Proceedings of the Americas conference on information systems. https://aisel.aisnet.org/amcis2020/ai_semantic_for_intelligent_info_systems/i_semantic_for_intelligent_info_systems/14ai_semantic_for_intelligent_info_systems/14

Sculley D, Holt G, Golovin D, Davydov E, Phillips T, Ebner D, Chaudhary V, Young M, Crespo J-F, Dennison D (2015) Hidden technical debt in machine learning systems. In: Cortes C et al (eds) Advances in Neural Information Processing Systems, vol 28. https://proceedings.neurips.cc/paper_files/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html

Shams R (2018) Developing machine learning products better and faster at startups. IEEE Eng Manag Rev 46:36–39. https://doi.org/10.1109/EMR.2018.2870669

Shimagaki J, Kamei Y, Ubayashi N, Hindle A (2018) Automatic topic classification of test cases using text mining at an Android smartphone vendor. In: Oivo M et al (eds) Proceedings of the 12th ACM/IEEE International Symposium on Empir Softw Eng and Measurement. ACM, New York

Someh I, Wixom B, Zutavern A (2020) Overcoming organizational obstacles to artificial intelligence value creation: propositions for research. In: Proceedings of the 53rd Hawaii International Conference on System Sciences

Sonnenberg C, vom Brocke J (2012) Evaluation patterns for design science research artefacts. In: Helfert M, Donnellan B (eds) Practical aspects of design science, vol 286. Springer, Heidelberg, pp 71–83

Venable J, Pries-Heje J, Baskerville R (2016) FEDS: a framework for evaluation in design science research. Eur J Inf Syst 25:77–89. https://doi.org/10.1057/ejis.2014.36

Wamba-Taguimdje S-L, Fosso Wamba S, Kala Kamdjoug JR, Tchatchouang Wanko CE (2020) Influence of artificial intelligence (AI) on firm performance: the business value of AI-based transformation projects. Bus Proc Manag J 26:1893–1924. https://doi.org/10.1108/BPMJ-10-2019-0411

Wan Z, Xia X, Lo D, Murphy GC (2020) How does machine learning change software development practices? IEEE Trans Softw Eng. https://doi.org/10.1109/TSE.2019.2937083

Washizaki H, Uchida H, Khomh F, Gueheneuc Y-G (2019) Studying software engineering patterns for designing machine learning systems. In: Proceedings of the 10th international workshop on Empir Softw Eng in Practice. IEEE, New York, pp 49–54

Weber M, Engert M, Schaffer N, Weking J, Krcmar H (2022) Organizational capabilities for ai implementation – Coping with inscrutability and data dependency in AI. Inf Syst Front. https://doi.org/10.1007/s10796-022-10297-y

Webster J, Watson RT (2002) Analyzing the past to prepare for the future: Writing a literature review. MIS Q 26(2):13–23

Wernerfelt B (1984) A resource-based view of the firm. Strateg Manag J 5:171–180. https://doi.org/10.1002/smj.4250050207

Whang SE, Lee J-G (2020) Data collection and quality challenges for deep learning. Proc VLDB Endowment 13:3429–3432. https://doi.org/10.14778/3415478.3415562

Yang Y, Zhan D-C, Fan Y, Jiang Y, Zhou Z-H (2017) Deep learning for fixed model reuse. In: Proceedings of the AAAI conference on artificial intelligence 31. https://doi.org/10.1609/aaai.v31i1.10855

Yao Y, Xiao Z, Wang B, Viswanath B, Zheng H, Zhao BY (2017) Complexity vs. performance: empirical analysis of machine learning as a service. In: Uhlig S, Maennel O (eds) Proceedings of the 2017 Internet Measurement Conference. ACM, New York, pp 384–397

Yi J, Zhang C, Wang W, Li C, Yan F (2020) Not all explorations are equal: harnessing heterogeneous profiling cost for efficient MLaaS training. In: Proceedings of the 34th International Parallel and Distributed Processing Symposium. IEEE, New York, pp 419–428

Yu J, Ke X, Xu F, Huang H (2020) Efficient architecture paradigm for deep learning inference as a service. In: Proceedings of the 39th international performance computing and communications conference. IEEE, New York, pp 1–8