What About the Data? A Mapping Study on Data Engineering for AI Systems

Petra Heck
p.heck@fontys.nl
Fontys University of Applied Sciences
Eindhoven, Netherlands

ABSTRACT

AI systems cannot exist without data. Now that AI models (data science and AI) have matured and are readily available to apply in practice, most organizations struggle with the data infrastructure to do so. There is a growing need for data engineers that know how to prepare data for AI systems or that can setup enterprise-wide data architectures for analytical projects. But until now, the data engineering part of AI engineering has not been getting much attention, in favor of discussing the modeling part. In this paper we aim to change this by perform a mapping study on data engineering for AI systems, i.e., *AI data engineering*. We found 25 relevant papers between January 2019 and June 2023, explaining AI data engineering activities. We identify which life cycle phases are covered, which technical solutions or architectures are proposed and which lessons learned are presented. We end by an overall discussion of the papers with implications for practitioners and researchers. This paper creates an overview of the body of knowledge on data engineering for AI. This overview is useful for practitioners to identify solutions and best practices as well as for researchers to identify gaps.

CCS CONCEPTS

• **Software and its engineering**; • **Information systems** → **Data structures**; • **Computing methodologies** → **Artificial intelligence**;

KEYWORDS

data-centric AI, AI engineering, data engineering, data architecture, DataOps, MLOps, data quality

1 INTRODUCTION

AI systems cannot exist without data [16]. To develop an AI system, data needs to be collected and prepared to train the AI model, see the DATA cycle in Figure 1. But also when the model is in production,

production data needs to be prepared to send it to the model and get back predictions. This data can be structured tabular data or unstructured data such as images, sound, text or video. In 2021, Andrew Ng coined the term "data-centric AI" for "the discipline of systematically engineering the data used to build an AI system"[1], i.e., *AI data engineering.*
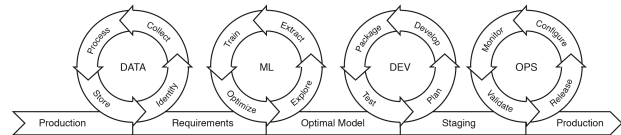
Figure 1: The AI engineering life cycle [12]

Reis and Housley [36] define data engineering as "the development, implementation, and maintenance, of systems and processes that take in raw data and produce high-quality, consistent information that supports downstream use cases, such as analysis and machine learning. Data engineering is the intersection of security, data management, DataOps, data architecture, orchestration and software engineering." Figure 2 shows the data engineering life cycle, consisting of Ingestion, Transformation, Serving and Storage. The definition of Reis and Housley is very broad, not just focused on one AI engineering project as is the life cycle in Figure 1, and also has a clear link to machine learning (thus AI).

Figure 2: The data engineering life cycle [36]

In the projects we do with industry we see that now AI models have matured and are readily available to apply in practice, most organizations struggle with the data infrastructure to do so. This can

---

[1]https://datacentricai.org/

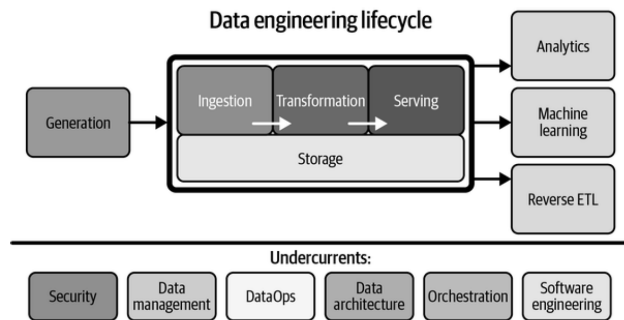be on a project level: How to retrain this pre-trained model with our own small dataset? How to create synthetic data? How to merge different data sources? How to store and version data? How to automate data pre-processing? But also on organization level: How to integrate all available data from different systems into a central "store"? How to protect sensitive data? How to deal with privacy issues? How to clean data? How to design an organization-wide data architecture that is fit for future AI developments? More and more organizations directly ask for data engineers as the main driver for their AI initiatives [27]. They struggle to find these employees, because software engineers have not been trained on "big data" or data architectures specifically and data scientists do not have the required software engineering skills. Both disciplines rather focus on the more attractive model development part of AI engineering.

Jarrahi et al. [21] also stress the importance of data-centric AI (DCAI). They state that "the nature of 'data work' itself is not necessarily new. However, over the years, the actual data work in AI projects comes mostly from individual initiatives, and/or from piecemeal and ad hoc efforts. A lack of attention to data excellence and quality of data has resulted in underwhelming outcomes for AI systems, particularly those deployed in high-stake domains such as medical diagnosis. DCAI magnifies the role of data throughout the AI life cycle and stretches its lifespan beyond the so-called 'preprocessing step' in model-centric AI."

One could argue that to account for true DCAI the AI engineering life cycle should also be extended with undercurrents such as "data architecture" and "DataOps", analogue to Figure 2. These are data-related activities that are not just relevant for one single AI engineering project, but for the entire organization, or for all projects being executed, or for all AI systems being maintained. This is in fact also what we see in the projects we do with industry. They know how to quickly get data for one machine learning experiment, but not how to set up a data architecture for enterprise-scale AI engineering.

This paper sets out to answer the question *"How to do data engineering for AI systems?"* by means of a mapping study. For this mapping study we formulated the following research questions:

- RQ1: Which data and AI engineering lifecycle phases are covered?
- RQ2: Which technical solutions (tools/frameworks/platforms) for AI data engineering are proposed?
- RQ3: Which architectures for AI data engineering are proposed?
- RQ4: What are lesson learned on AI data engineering?

The mapping study identified 25 papers that explain data engineering activities, tools, frameworks or architectures. By categorizing them and summarizing their solutions and lesson learned, the paper creates an overview of the body of knowledge on data engineering for AI. This overview is useful for practitioners to identify solutions as well as for researchers to identify gaps.

The remainder of the paper is organized as follows. Section 2 provides related work on data engineering for AI systems. Section 3 explains the method used for the mapping study and how the 25 resulting papers were selected. Section 4 classifies the 25 papers according to their meta-data, the type of solution they discuss and the scope of the data engineering activities explained. Section 5 analyzes each of the four subquestions (RQ1 till RQ4). Section 6 discusses the overall research question and implications for practitioners and researchers. The paper concludes with main findings and future work.

2 RELATED WORK

This section describes related work on data engineering within AI engineering research.

Amershi et al. [3] is one of the first AI engineering case studies to appear. The paper includes a machine learning workflow including data-oriented steps. It describes data engineering challenges at Microsoft but not really explains solutions. The paper shows that already in 2019, data management and data discovery on a project-level was a challenge for AI engineers. The case study does not describe data engineering challenges on enterprise level.

Serban et al. [41] identified a set of best practices for AI engineering from existing literature, including five best practices on data:

1. Use sanity checks for all external data sources
2. Check that input data is complete, balanced and well distributed
3. Write reusable scripts for data cleaning and merging
4. Ensure data labelling is performed in a strictly controlled process
5. Make data sets available on shared infrastructure (private or public)

More information on these best practices (and how to implement them) might be taken from the papers that list them. Since Serban et al. is a meta-research it is not included in our mapping study.

Several authors [7, 13, 19, 28, 30, 32, 39, 42] describe data challenges for AI systems, without offering explicit solutions. Each of those papers does however stress the importance of data engineering for AI systems. Bosch et al. [7] present DataOps as part of the AI engineering research agenda as a "a significant opportunity to reduce ... overhead by generating, distributing and storing data smarter in the development process". Sambasivan et al. [39] state that "Data quality carries an elevated significance in high-stakes AI due to its heightened downstream impact, impacting predictions like cancer detection, wildlife poaching, and loan allocations. Paradoxically, data is the most under-valued and de-glamorised aspect of AI.". This paper complements previous work with an overview of actionable solutions.

3 PAPER SELECTION

To select papers that answer our research questions on data engineering for AI we used the process described by Kitchenham and Charters [22]:

1. Define inclusion and exclusion criteria
2. Design query string
3. Identify databases and other sources to search
4. Select relevant papers based on title and abstract
5. Select relevant papers based on full text
6. Extend result set based on citations
7. Classify resulting papers

What About the Data? A Mapping Study on Data Engineering for AI Systems

CAIN 2024, April 14–15, 2024, Lisbon, Portugal

As a final step, we coded [38] the resulting papers to answer the research questions. The following paragraphs describe each of the steps in detail. The classification step is described in Section 4.

3.1 Inclusion and Exclusion Criteria

We used four *inclusion* criteria. Three of them are on metadata of the paper: the paper should be in English, peer-reviewed and from 2019 or later. We choose 2019 because the AI engineering publications are all from that year or later, e.g., the seminal Microsoft case study by Amershi et al. [4]. The fourth inclusion criterion is on the content of the paper. The paper should explain data engineering activities, best practices, tools, frameworks or data architectures in the context of AI engineering (building production-ready AI systems). We *exclude* books, theses (bachelor, master and PhD.) and meta-research such as systematic literature reviews or mapping studies. Furthermore we excluded papers that only mention challenges, without providing solutions.

3.2 Query String

As we are looking for data engineering for AI it makes sense to include both these terms in the query. We choose to use the more specific term "AI engineering" since we are interested in the software engineering view on AI. We also included "data requirements" and "data architecture" to surface papers that describe data artefacts (and thus possibly data activities) without mentioning "data engineering".

> **(data engineering || data requirements || data architecture) & AI engineering**

3.3 Source Selection and Query Execution

We selected five digital databases which index software engineering venues plus Google Scholar. Furthermore we chose to try out Elicit, an AI-powered research assistant. Elicit uses language models to extract data from and summarize research papers. A search in Elicit is based on a question that you are trying to answer, not on a query string. We directly asked Elicit to come up with papers that answer the question "How to do data engineering for AI engineering". Elicit generates papers in batches, and we stopped when a newly generated batch did not contain relevant titles anymore. We executed the query string on each digital database (June 2023), resulting in 259 unique items in total[2]:

- Google Scholar (scholar.google.com) 182 items
- IEEE Xplore (ieeexplore.ieee.org) 5 items, 0 unique items
- ACM Digital Library (dl.acm.org) 8 items, 1 unique item
- Scopus (scopus.com) 3 items, 0 unique items
- DBLP (dblp.org) 16 items, 13 unique items
- ScienceDirect (sciencedirect.com) 7 items, 2 unique items
- Elicit (elicit.org) 64 items, 61 unique

3.4 Paper Selection and Citation Snowballing

For the 259 unique items we determined whether it is a peer-reviewed paper (no book, thesis or meta-research) that describes data engineering for AI.

[2]Dataset available online at https://dx.doi.org/10.13140/RG.2.2.17748.78725

Table 1: Excluded items

	# Items
Book, thesis or meta-research	52
Not peer-reviewed	53
No data engineering activities	28
No AI engineering - Software engineering	3
No AI engineering - Data Science	53
No AI engineering - AI governance	10
No AI engineering - Education	5
No AI engineering - Infrastructure, Hardware	14
No AI engineering - AI4SE	5
Only mentions, no explanation	7
Only challenges, no solutions	10
Total excluded	**240**

Table 2: Filtering publications on data engineering for AI

	Query	Inclusion	References
Google Scholar	182	11	15
ScienceDirect	2	0	0
ACM DL	1	0	0
DBLP	13	1	1
Elicit	61	7	9
TOTAL	**259**	**19**	**25**

In total we excluded 240 items. With those items we performed card sorting on the reason for exclusion, resulting in eleven categories, see Table 1. The card sorting serves as a soundness check of the exclusion process (did we exclude for the right reason?).

The selection process resulted in 19 (= 259 - 240) papers.

With the 40 papers that were initially included based on title and abstract, we also performed snowballing according to the guidelines provided by [48]. We checked all the references in the 40 papers, but also checked all citations of these 40 papers with Google Scholar. We repeated this snowballing process until no new papers were added. The complete snowballing process has added 6 new papers to this final set, resulting in 25 papers in total (see Table 2).

4 PAPER CLASSIFICATION

The resulting set of 25 papers was classified according to different dimensions. This section explains the different classifications.

4.1 Classification by Meta-Data

To indicate the background of the selected papers we classified them by the following meta-data.

(1) Author affiliation (Aff. = University, Company, Public Organization, Research Center)
(2) Author country;
(3) Year of publication;
(4) Number of pages (#p);
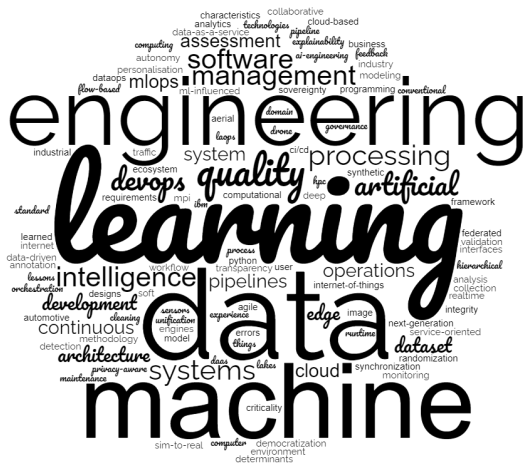(5) Number of citations (#cit.);
(6) Keywords;

Figure 3: Word cloud of the keywords from the 25 selected papers

(7) Focus on software engineering (SE) or focus on data science (DS).

4.2 Classification by Scope of Data Engineering

To indicate the scope of data engineering activities explained in each paper we classified them according to the following categories.

(1) Data pipeline for training AI systems (TP);
(2) Data pipeline for serving data to AI systems in production (PP);
(3) System-wide AI data architecture (DA);
(4) Enterprise-wide AI data architecture (EDA);

4.3 Classification by Type of Data Engineering Solution

To indicate what type of guidance on data engineering for AI the paper provides we classified them according to the following categories.

(1) Technical solution (tool, platform, library, etc.);
(2) Architecture;
(3) Best practices;
(4) Case study;

To provide better guidance for practitioners, we also indicate for each paper which of the following empirical validation of the provided solutions was done.

(1) Case study in industry;
(2) Validation within author's company;
(3) Experiments;
(4) Demo applications or implementations;
(5) No empirical validation.

4.4 Classified Papers

Table 3 shows how each of the 25 selected papers classifies on metadata, scope, and solution type.

What stands out in this table is that the set of 25 papers comes from a mix of industry and academics, with only 9 being purely academic. Most papers (15 out of 25) come from European countries, with a large number from Germany (6) and Sweden (3). Note that the 3 papers from Sweden seem to come from the same research group. Most papers (17) focus on the software engineering perspective, which is not surprising, since we specifically selected on "AI engineering". With respect to the types of solutions discussed in the papers, all of them are well represented. Furthermore, most papers contain some form of empirical evaluation of their solutions.

With respect to the scope of activities explained in the paper, eleven focus on the training pipeline (TP) and seven focus on the production pipeline (PP). Only four papers focus on system-level data engineering and only three on enterprise-level data engineering. This means most papers take quite a narrow definition of data engineering: setting up a data pipeline for a machine learning model.

With respect to the keywords, the list is long and quite diverse, see the word cloud in Figure 3. Papers DE1, DE2 and DE14 are without keywords, the other papers have selected between three and ten (DE16) keywords. DE13, DE15, DE16, DE17, DE21 and DE24 do not mention data in their keywords, but they do mention AI or machine learning. The other papers all selected keywords that include "data":

- data processing: DE3, DE4
- data engineering: DE5, DE12
- DataOps: DE6
- data pipelines: DE3 (pipeline), DE6, DE9
- data technologies: DE6
- data management, data democratization, data governance, data ecosystem: DE7
- data quality: DE8, DE23
- data errors: DE8
- data validation: DE8
- data transparency: DE10
- data cleaning: DE10
- data sovereignty: DE11
- data-driven development: DE18
- data integrity: DE19
- hierarchical dataset, dataset design: DE20
- data collection standard, data synchronization: DE20
- Data-as-a-Service, DaaS: DE22
- synthetic data: DE25

This list of keywords already shows that terminology within the 25 papers is not standardized and that many different aspects of data engineering for AI are being considered. We will analyze this in more detail in the next section, related to the research questions.

5 DATA ENGINEERING FOR AI SYSTEMS

In this section we discuss our findings related to each of the four research questions. To answer the research questions, the full text of each paper was coded according to 1) the life cycle phases that

Table 3: Papers on data engineering for AI

ID	Ref.	Year	Affiliation	Country	#p	#cit.	SE/DS	Scope	Type	Validation
DE1	[8]	2019	Company (Google)	USA	12	238	SE	PP	Technical solution	Case study
DE2	[11]	2019	University	Germany	12	23	DS	PP	Technical solution	Experiments
DE3	[15]	2019	University	USA	4	4	SE	PP	Technical solution	Demo
DE4	[49]	2019	Company (Fujitsu)	Japan	8	41	SE	DA	Architecture	Demo
DE5	[1]	2020	University	USA, Sri Lanka	9	5	SE	TP	Technical solution	Experiments
DE6	[29]	2020	Univ. + Comp. (Ericsson)	Sweden	10	66	SE	EDA	Best practices	Company
DE7	[16]	2021	Company (Robert Bosch)	Germany	9	36	SE	EDA	Architecture	Company
DE8	[24]	2021	Univ. + Comp. (Ericsson)	Sweden	10	13	SE	PP	Best practices	Company
DE9	[35]	2021	Univ. + Research C.	Sweden	10	1	SE	PP	Best practices	Case study
DE10	[45]	2021	Research Center	Qatar	9	2	DS	TP	Architecture	No
DE11	[2]	2022	Research C. + Univ.	Germany, Spain	12	0	SE	TP	Case study	Case study
DE12	[9]	2022	Univ. + Public Org.	USA	6	1	DS	TP	Case study	Case study
DE13	[10]	2022	University	Australia	6	1	SE	TP	Best practices	No
DE14	[14]	2022	Univ. + Research C.	Austria	11	11	SE	PP	Technical solution	Experiment
DE15	[17]	2022	Research Center	Germany	10	5	SE	DA	Best practices	No
DE16	[31]	2022	University	Finland	8	0	SE	DA	Case study	Case study
DE17	[33]	2022	University	UK	11	3	SE	DA	Architecture	Demo
DE18	[34]	2022	Research Center	Germany	8	1	SE	TP	Best practices	Demo
DE19	[40]	2022	Research Center	Norway, Spain	7	2	SE	EDA	Architecture	Case study
DE20	[46]	2022	Research Center	China	6	1	DS	TP	Architecture	Demo
DE21	[47]	2022	University	Austria	11	4	SE	TP	Architecture	No
DE22	[6]	2023	University	Italy	19	0	DS	PP	Architecture	Demo
DE23	[20]	2022	University	India, Vietnam	14	1	DS	TP	Technical solution	Demo
DE24	[23]	2023	Univ. + Comp. (IBM)	Germany	10	49	SE	TP	Architecture	Interview
DE25	[37]	2023	Univ. + Comp. (Microsoft)	USA	17	0	DS	TP	Technical solution	Experiments

SE=software eng., DE=data science, TP/PP=training/production pipeline, DA/EDA=system/enterprise-wide data architecture

are described; 2) the technical solutions that are described; 3) the architecture pictures it contains; 4) the lessons learned it contains.

5.1 RQ1: Which Data and AI Engineering Life Cycle Phases Are Covered?

To map the 25 selected papers to life cycle phases, we coded them with the AI engineering life cycle phases from Figure 1 (DATA, ML, DEV and OPS) and the data engineering life cycle phases from Figure 2 (Generation, Ingestion, Transformation, Serving). When the paper does not focus on one or more specific life cycle phase(s), we coded it with "All". Table 4 shows the division of the papers over the life cycle phases.

Conclusion. Not surprisingly the majority of papers cover at least the DATA phase of the AI engineering life cycle. But eight of the papers focus more on the DEV and/or OPS part of the AI engineering life cycle. Out of them, five papers specifically focus on data validation in production (the Serving phase of the Data Engineering life cycle).

5.2 RQ2: Which Technical Solutions for AI Data Engineering Are Proposed?

As can be seen in Table 3, seven of the papers discuss technical solutions for AI data engineering. In this section we discuss each of the seven proposed solutions in more detail.

[DE1] Breck et al. [8] present a "data validation system that is designed to *detect anomalies specifically in data fed into machine*

Table 4: RQ1: Which life cycle phases are covered?

AI Eng.	Data Eng.	Papers
DATA	Generation	DE18, DE20, DE23, DE25
DATA	Transformation	DE10, DE12
DATA	All	DE6, DE21
DATA+ML	Transformation	DE3
DATA+ML+DEV	All	DE17
DEV	All	DE4
DEV+OPS	All	DE9, DE19
OPS	Serving	DE1, DE2, DE8, DE14, DE22
All	All	DE5, DE7, DE11, DE13, DE15, DE16, DE24

learning pipelines. This system is deployed in production as an integral part of TFX – an end-to-end machine learning platform at Google." They discuss the challenges they faced in developing the system and the techniques they used to address them, including design choices that were made. They also present three case studies at Google to illustrate the benefits of the data validation system in production.

[DE2] Derakshan et al. [11] propose a "*platform for continuously training deployed machine learning models* and pipelines that adapts to the changes in the incoming data." Their platform uses techniques such as proactive training, online statistics computation and dynamic materialization to reduce (re)training and deployment

costs. They include evidence from experiments, with two different machine learning pipelines.

[DE3] Frost et al. present "AI Pro, an open-source framework for *data processing* with Artificial Intelligence (AI) models." With AI Pro users can generate a data pipeline from a configuration file through a user friendly web interface. For advanced users and core developers, there is a command line interface for in-depth operations with finer-grained control. They demonstrate AI Pro with two demo scenarios.

[DE5] Abeykoon et al. [1] developed a "*high performance Python API* with a C++ core to represent data as a table and provide distributed data operations." Their PyCylon solution bridges ETL pipelines in Python (as mostly used by data scientists) with high performance compute kernels in C++. They conducted experiments to proof the performance of PyCylon.

[DE14] Foidl et al. [14] collected a catalogue of 36 "data smells" in a multi-vocal literature review and implemented *tool support to detect these data smells*. They applied the tools to 246 Kaggle datasets to evaluate them. As opposed to the data anomalies detected by the system of [DE1] presented above, these data smells are broader, since smells also include "*potential* data quality issues".

[DE23] Jariwala et al. [20] demonstrate the use of the *IBM Data Quality for AI Toolkit* to check training data in a machine learning setting. They include a workflow how to call the IBM API and show the results of several included data quality metrics on open source datasets.

[DE25] Sabet et al. [37] introduce "a scalable Aerial *Synthetic Data Augmentation* (ASDA) framework tailored to aerial autonomy applications." They demonstrate the ASDA framework by generating data for landing pad detection in the Seattle simulation scene. Although this is a very specific technical solution, the usage of synthetic datasets is of course not limited to aerial autonomy applications.

Conclusion. The selected papers contain a diverse set of technical solutions, ranging from synthetic data generation (DE25), through data validation tools (DE1, DE14, DE23), through data processing frameworks (DE3, DE5), to deployment platforms (DE2). In that way the solutions presented together cover the complete AI engineering and data engineering life cycles, although most solutions focus on one single life cycle phase (see Table 4). The exception is [DE5] that does not cover one specific life cycle phase and is the only paper that covers the DEV phase of AI engineering.

5.3 RQ3: Which Architectures for AI Data Engineering Are Proposed?

As can be seen in Table 3, nine of the papers discuss architectures for AI data engineering. In this section we discuss each of the nine architectures in more detail.

[DE4] Yokoyama [49] propose a *multi-layer architectural pattern for machine learning systems* that separates the business logic from the inference engine and data processing. Furthermore, it separates the user interface from the data collection and the data lake from the database. They demonstrate their architectural pattern by designing a chatbot system.

[DE7] Gröger [16] calls for a *data ecosystem* for industrial enterprises, see Figure 4. That ecosystem contains a specific role for data

engineers and data engineering as part of the data democratization challenge: "making all kinds of data available for AI for all kinds of end users across the entire enterprise". Gröger suggests to address this challenge with an enterprise data catalog that provides comprehensive metadata management across all data lakes and other data sources. This would enable self-service use of data.

[DE10] Thirumuruganathan et al. [45] present a *reference architecture for automated annotations of data.* They describe the key components of this system architecture. Implementing a proof-of-concept remains future work.

[DE17] Paleyes et al. [33] propose "*Flow-Based Programming* as a paradigm for creating Data Oriented Architecture (DOA) applications." They compared the flow-based programming (FBP) paradigm to the Service-Oriented Paradigm (SOA) by implementing four data-driven applications in both paradigms and measuring evolution of the codebase through pre-defined metrics.

[DE19] Sen et al. [40] devised a "*de-centralized edge-to-cloud architecture*" with machine learning pipelines for erroneous data repair and detection of deviations in sensor data. They analyze their proposed architecture in two different industrial case studies.

[DE20] Wang et al. [46] implement a *hierarchical dataset* with unified annotation rules. They use one example scenario to compare a hierarchical dataset created from three single datasets with the original single-source dataset.

[DE21] Warnett and Zdun [47] list *architectural design decisions (ADDs) for the machine learning workflow* from a gray literature study. Their replication package contains in an ADD model with UML diagrams of all ADDs and their relations.

[DE22] Azimi and Pahl [6] present a *layered Data-as-a-Service (DaaS) quality management architecture.* Their framework focuses on input data quality and links it to machine learned data service quality. They demonstrate their framework with a traffic management use case.

[DE24] Kreuzberger et al. [23] depict an "*end-to-end MLOps architecture and workflow* with functional components and roles". The workflow contains a separate data engineering zone and data(Ops) engineer is a separate role (apart from e.g., software engineer, data scientist or even ML engineer). According to them a data engineer "builds up and manages data and feature engineering pipelines" and "ensures proper data ingestion to the databases of the feature store system". This indicates that their architecture/workflow has the scope of one single ML project.

Conclusion. Most architectures presented focus on (parts of a) system architecture (DE4, DE10, DE17, DE20, DE22) or the ML pipeline (DE21, DE24). Only papers DE7 and DE19, contain a diagram for enterprise-wide data architectures, of which DE19 focuses on IoT data only. The data ecosystem from DE7 contains a comprehensive overview of this enterprise data landscape, including IoT data sources, see Figure 4.

5.4 RQ4: What are Lessons Learned on AI Data Engineering?

As can be seen in Table 3, nine of the papers discuss case studies or best practices for AI data engineering. In this section we discuss each of the nine papers in more detail.
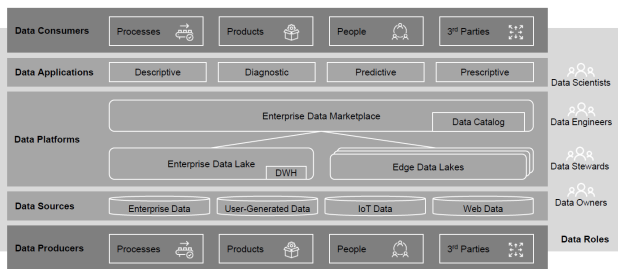
What About the Data? A Mapping Study on Data Engineering for AI Systems

CAIN 2024, April 14–15, 2024, Lisbon, Portugal

Figure 4: A data ecosystem for industrial enterprises [16]

[DE6] Raj et al. [29] derive a definition of DataOps from literature including the main components identified. According to their analysis, DataOps can be defined as "an approach that accelerates the delivery of high quality results by automation and orchestration of data life cycle stages." They describe eight DataOps use cases at Ericsson and derived a five stage *DataOps evolution* from them. For each stage they define requirements, which can also be read as best practices.

[DE8] Lwakatare et al. [25] conducted action research at Ericsson for *training data validation*. Based on their research, they propose a data validation framework that considers multiple levels of data validation checks (feature, dataset, cross-dataset, data stream) and provision of feedback. Their research also identified three best practices: 1) define data quality tests at all four levels, 2) provide actionable feedback and suggest mitigation strategies, 3) treat data errors with similar rigor as code.

[DE9] Raj et al. [35] conducted a *multiple-case study into six data pipelines* from commercial software-intensive systems at three companies. They describe and explain seven determinants for data pipelines: 1) Big Data, 2) data preprocessing, 3) data quality, 4) data storage requirements, 5) data pipeline elements, 6) performance efficiency, 7) continuous monitoring and fault detection. These determinants are important factors to consider when implementing a data pipeline.

[DE11] Altendeiterung et al. [2] conducted a twelve-month *action research at Mondragon on data-sovereign AI pipelines*. They explicitly list ten lessons learned: 1) need for data traceability, 2) need for an independent trustee, 3) need for quality-driven data sharing, 4) need for a data catalog, 5) need for real-time support, 6) need for a separation of control and data plane, 7) need for access and usage control enforcement, 8) need for standardization, 9) need for a common definition of user roles, 10) need for a trusted and secure deployment environment.

[DE12] Chattopadhyay et al. [9] describe an experiment where they used six different ways to *transform a rainfall dataset from a five-minute span to a single value* (baseline, mean, median, mode, maximum, minimum). They call this choice a "data engineering decision" and argue that the choice made impacts model results quantitatively and qualitatively.

[DE13] Cheng and Long [10] propose *Federated Learning Operations (FLOps)* as a methodology for developing cross-silo federated learning systems. They describe a life cycle with three phases and fourteen activities (one of which is data engineering). They also provide three best practices: 1) metadata engineering to create Data

Interface Abstractions (DIAs) for models, 2) dual deployment of models and DIAs, 3) check points between phases.

[DE15] Hasterok and Stompe [17] present the *Process Model for AI Systems Engineering (PAISE®)* as an extension of ISO/IEC 15288 with AI engineering. PAISE® specifies its own procedures for ML component development and data provisioning. The data provisioning process facilitates the development of datasets as separate components in the system. It also is a feedback loop, where datasets can be updated based on the monitoring of the system.

[DE16] Niemelä et al. [31] propose an *architecture for a learning analytics system (LAOps)* in which they combine MLOps and privacy-aware cryptographic data storage. They describe the LAOps implementation they currently have at Tampere University.

[DE18] Petersen et al. [34] introduce a *data-driven workflow for developing qualitative datasets in automative systems engineering*. They showcase this process by "curating a data pool consisting of different available data sources that have to be integrated to cover as many driving situations as possible."

Conclusion. The lessons learned from the above nine papers are quite diverse and with different scope. DE6 about DataOps is probably the most broad one together with DE15 about PAISE® (process model for AI engineering), whereas DE12 describes just one data wrangling step in one single project. Depending on the context of the AI engineering project, different papers apply: DE11 and DE13 for federated learning, DE16 for learning analytics and DE18 for automotive systems engineering. DE8 and DE9 focus on one specific part of the life cycle: automated training data validation (DE8) and data pipelines for serving data to machine learning models in production (DE9).

6 DISCUSSION

This section discusses the findings in light of the overall research question "*How to do data engineering for AI systems?*"

6.1 Threats to Validity

Most threats to validity in such a mapping study relate to researcher bias in selecting and coding papers. We mitigated this by 1) following the guidelines suggested by Kitchenham and Charters [22], Saldãna [38] and Wohlin [48]; 2) documenting and reviewing all steps we made; 3) using existing life cycle models and definitions for the coding; 4) making available the entire dataset, including selection and coding for other researchers to validate our results.

Note that because we specifically searched for "AI engineering" in the query string, we might have missed papers that refer to AI engineering with other wording. We mitigated this by also searching with AI-based tool Elicit (61 new papers of which 7 were included in the result set) as well as by snowballing from the other selected papers.

6.2 Defining Data Engineering for AI Systems

Before we can answer the question "How to do AI data engineering?", we must first answer the question "What is AI data engineering?". All 25 selected papers take a different angle. Data engineering might refer to one single task or step in an AI engineering project, a discipline within software engineering or data science, or an enterprise-wide competency, see also Table 2 (TP, PP, DA or EDA).

Out of all papers, DE12 takes the most narrow view on data engineering. Chattopadhyaya et al. [9] write about data engineering decisions for AI-based applications. They use this rather umbrella term to indicate the decision how they should convert a timestamp dataset into an interval-based dataset: take the mean, the mode, the maximum or the minimum for that interval.

Only 4 out of 25 papers actually define what they mean by data engineering. Paper DE5 by Abeykoon et al. [1] defines data engineering as "The complex process of transforming raw data to a form suitable for analytics". Paper DE6 by Raj et al. [29] has a rather fuzzy definition of data engineering as a step that "performs two different operations at the high level, which include data collection and data ingestion". Paper DE7 by Gröger [16] defines data engineering as "modelling, integrating and cleansing of data." Paper DE13 by Cheng and Long [10] says "the raw data in each entity is extracted, transformed, and prepared for model training". These definitions are all also much more narrow than the definitions from Andrew Ng and Reis and Housley as given in the introduction.

To get a complete picture of data engineering for AI systems, one should combine the life cycles in Figure 1 and Figure 2. In that way, one has both a picture of how AI engineering connects to data as well as how data engineering connects to AI. We did not find any comprehensive work that already does this and only three papers that take a similar enterprise-level view on data engineering (DE6, DE7 and DE19).

6.3 Implications for Practitioners

The mapping of the selected papers to life cycle phases (RQ1, see section 5.1) and type of solutions they provide (RQ2 till RQ4, see section 5.2 till 5.4) provides guidance to practitioners which solutions to select for which project or activity. Running our mapping study, we also had the following observations on AI data engineering that could be useful for practitioners.

Big Data. During snowballing, we excluded a number of papers on data engineering for Big Data, that did not have an explicit reference to AI systems. However, those papers might contain valuable solutions for both researchers and practitioners that also holds for AI systems, as these systems are mostly Big Data systems as well. That kind of analysis was out of scope for this paper and might be a topic for a future mapping study: "How to do data engineering for Big Data?"

Data quality. A number of papers relate to data quality or data validation. There might be an interesting body of knowledge (and tools) on those topics that was out of scope for our mapping study. Zhang et al. [50] list several methods for data testing in their survey on machine learning testing. The concept of data smells [14, 43] is also important to consider in this context as these indicate data quality issues that might lead to machine learning problems, different from data errors (see DE14 [14]).

Grey literature. In our mapping study we included only peer-reviewed papers. However, we found a number of other interesting resources: 1) The original post of Figure 1 [12] that already defined the DATA cycle as in fact being a DataOps process, 2) the blogs and whitepapers on DataOps referenced in DE6 [29], 3) the book "Fundamentals of Data Engineering" [36] from which we borrowed

Figure 2, 4) other books such as "Data Fabric and Data Mesh Approaches with AI" [18] and "Practical DataOps: Delivering agile data science at scale" [5]. This means that practitioners should definitely consider sources from grey literature on DataOps and modern data architectures (such as data meshes and data fabrics).

Open source tooling. The open source tooling landscape for data is becoming bigger and bigger, see also the online blog post "The State of Data Engineering"[3]. These kind of tools are necessary to achieve higher levels of DataOps, see DE6 [29].

Data spaces. Some AI data engineering projects require data sharing between different organizations or entities. To support federated learning and data sovereignty several technical solutions such as Gaia-X, FIWARE and the International Data Space have been built up. Paper DE11 by Altendeiterung et al. [2] investigates how to integrate such solutions with AI pipelines, but we recommend to also keep an eye on the evolution of the data space solutions, since they are fairly new.

Domain-specific data engineering. Two domains that stood out in our mapping study are data engineering for Internet-of-Things (IoT) and data engineering for automotive. There might be more literature or guidance on AI data engineering if one dives into a specific domain.

Synthetic data. Paper DE25 by Sabet et al. [37] describes synthetic data generation. That is a topic that might not be relevant for all AI engineering projects, but if it is, we would like to point out that there is a whole body of knowledge (and tools) about synthetic data generation specifically that can be looked into by practitioners.

6.4 Implications for Researchers

The mapping of the selected papers to life cycle phases (RQ1, see section 5.1) and type of solutions they provide (RQ2 till RQ4, see section 5.2 till 5.4) provides guidance to researchers to see what is already there and what is still missing. Running our mapping study, we also had the following observations on AI data engineering that could be useful for researchers.

Data engineering. Tebernum et al. [44] developed a "data engineering reference model (DERM) which outlines the important building blocks for handling data along the data lifecycle." They aim to bridge between data engineers and software engineers by providing a common ground for engineering data-intensive applications. They view data engineering as a sub-discipline of data science ("preparing data for data scientists"). The AI engineering research community could benefit from integrating with the data science research community on the data engineering topic. The DERM presented by Tebernum et al. could serve as a common ground also for this purpose. Tebernum et al. also show ample opportunities for future data engineering research.

DataOps. Paper DE6 by Raj et al. [29] points to DataOps as an overall process to automate and orchestrate data life cycle stages. The large amount of references to grey literature they used indicates that DataOps is not receiving enough attention in research yet. In line with the evolution from DevOps to MLOps, there is also a

---

[3]https://lakefs.io/blog/the-state-of-data-engineering-2023/

What About the Data? A Mapping Study on Data Engineering for AI Systems

CAIN 2024, April 14–15, 2024, Lisbon, Portugal

need to evolve DataOps as a separate research field. This means that researchers should not consider data engineering as a single activity in a machine learning project, but as an approach that accelerates the entire enterprise data life cycle.

Data-Oriented Architecture (DOA). Paper DE17 by Paleyes et al. [33] introduces the term DOA as opposed to Service-Oriented Architecture (SOA). DOA seems an interesting paradigm for AI data engineering, but a quick search in Google Scholar only yields 23 articles that relate to DOA. More research is needed to establish if and how DOAs can be used to solve AI engineering challenges.

Enterprise-wide data architectures. The previous section points practitioners to books about data fabrics and data meshes. These are new concepts for managing data within an enterprise. The question how to effectively engineer AI systems making use of these concepts, remains open.

Data spaces and federated learning. The same holds for data spaces and federated learning. Paper DE11 by Altendeiterung et al. [2] investigates how to combine IDS with AI pipelines, but more research is needed on combining data spaces with federated learning.

Production data. Most papers focus on data engineering for training data. Now that AI has matured and more and more projects go into production, we need more background on how to engineer production data pipelines, how to validate and monitor production data, and how to set up enterprise-wide production data architectures.

Open source tooling. Open source tooling could also be a vehicle for researchers to transfer results to practitioners. In the area of AI engineering, open source tooling is wide-spread in industry, so researchers can easily integrate their data engineering solutions.

Knowledge engineering. Mattioli et al. [26] contrasts data-driven AI and knowledge-driven AI and argue that a hybrid approach is needed to build trustworthy AI systems. They point to the discipline of knowledge engineering, separate from data engineering. According to them, "knowledge engineering (KE) is the process of understanding and then representing human knowledge in data structures, semantic models (conceptual diagram of the data as it relates to the real world) and heuristics." In that way, it complements data engineering as it creates the data structures that data engineering deals with. This paper specifically focuses on data-driven AI engineering, but it is an interesting question how to combine this with knowledge-driven AI engineering.

Systems engineering. We focused our mapping study on AI engineering, thinking it to be a discipline within software engineering. However, DE15 about PAISE® and DE18 about a data-driven workflow describe AI engineering as a discipline within systems engineering. Then the data engineering part is not about enterprise-wide data, but about data within one system (e.g., device or machine). The AI engineering research community could benefit from integrating with the AI system engineering research community, as they might run into similar data-related challenges.

7 CONCLUSION

In this paper we created an overview of existing literature on data engineering for AI systems from an AI engineering perspective.

We found that most papers focus on engineering training or production pipelines for AI systems, but that they lack overall data architecture guidance for AI systems or the AI-driven enterprise. For software engineers and software engineering researchers this means that after DevOps and MLOps, now DataOps (and the integration between the three) is a new important topic to address. There is a strong need for frameworks, best practices, but also open source tools to support practitioners in implementing them. This paper provides a first overview of what is already there.

Future work remains to update the analysis, preferably also include available grey literature and books, and learn from case studies what is missing in practice. Our ultimate goal is to develop a data engineering toolbox for AI engineers, that includes both tooling to support project-level data pipelines as well as enterprise level data architectures. And, most importantly, an integrated data engineering and AI engineering approach.

ACKNOWLEDGMENTS

REFERENCES

[1] Vibhatha Abeykoon, Niranda Perera, Chathura Widanage, Supun Kamburugamuve, Thejaka Amila Kanewala, Hasara Maithree, Pulasthi Wickramasinghe, Ahmet Uyar, and Geoffrey Fox. 2020. Data engineering for hpc with python. In *2020 IEEE/ACM 9th Workshop on Python for High-Performance and Scientific Computing (PyHPC)*. IEEE, 13–21.

[2] Marcel Altendeitering, Julia Pampus, Felix Larrinaga, Jon Legaristi, and Falk Howar. 2022. Data Sovereignty for AI Pipelines: Lessons Learned from an Industrial Project at Mondragon Corporation. In *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI* (Pittsburgh, Pennsylvania) *(CAIN '22)*. Association for Computing Machinery, New York, NY, USA, 193–204. https://doi.org/10.1145/3522664.3528593

[3] Sa Amershi. 2019. Software Engineering for Machine Learning Applications. *Icse* 2020 (2019), 1–10. https://fontysblogt.nl/software-engineering-for-machine-learning-applications/

[4] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. 2019. Software engineering for machine learning: A case study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, 291–300.

[5] Harvinder Atwal. 2019. *Practical DataOps: Delivering agile data science at scale.* Springer.

[6] Shelernaz Azimi and Claus Pahl. 2021. AI Quality Engineering for Machine Learning Based IoT Data Processing. In *International Conference on Cloud Computing and Services Science.* Springer, 69–87.

[7] Jan Bosch, Helena Holmström Olsson, and Ivica Crnkovic. 2021. Engineering AI systems: A research agenda. *Artificial Intelligence Paradigms for Smart Cyber-Physical Systems* (2021), 1–19.

[8] Eric Breck, Neoklis Polyzotis, Sudip Roy, Steven Whang, and Martin Zinkevich. 2019. Data Validation for Machine Learning.. In *MLSys*.

[9] Aurek Chattopadhyay, Matthew Van Doren, Reese Johnson, and Nan Niu. 2022. On the Role of Data Engineering Decisions in AI-Based Applications.. In *REFSQ Workshops*.

[10] Qi Cheng and Guodong Long. 2022. Federated Learning Operations (FLOps): Challenges, Lifecycle and Approaches. In *2022 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*. 12–17. https://doi.org/10.1109/TAAI57707.2022.00012

[11] Behrouz Derakhshan, Alireza Rezaei Mahdiraji, Tilmann Rabl, and Volker Markl. 2019. Continuous Deployment of Machine Learning Pipelines.. In *EDBT*. 397–408.

[12] Danny Farah. 2020. The Modern MLOps Blueprint. online. https://medium.com/slalom-data-analytics/the-modern-mlops-blueprint-c8322af69d21

[13] Lukas Fischer, Lisa Ehrlinger, Verena Geist, Rudolf Ramler, Florian Sobiezky, Werner Zellinger, David Brunner, Mohit Kumar, and Bernhard Moser. 2020. AI system engineering—key challenges and lessons learned. *Machine Learning and Knowledge Extraction* 3, 1 (2020), 56–83.

[14] Harald Foidl, Michael Felderer, and Rudolf Ramler. 2022. Data smells: categories, causes and consequences, and detection of suspicious data in AI-based systems. In *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*. 229–239.

[15] Richie Frost, Debjyoti Paul, and Feifei Li. 2019. AI pro: Data processing framework for AI models. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 1980–1983.

[16] Christoph Gröger. 2021. There is no AI without data. *Commun. ACM* 64, 11 (2021), 98–108.

[17] Constanze Hasterok and Janina Stompe. 2022. PAISE®–process model for AI systems engineering. *at-Automatisierungstechnik* 70, 9 (2022), 777–786.

[18] Eberhard Hechler, Maryela Weihrauch, and Yan Wu. 2023. Data Fabric and Data Mesh for the AI Lifecycle. In *Data Fabric and Data Mesh Approaches with AI: A Guide to AI-based Data Cataloging, Governance, Integration, Orchestration, and Consumption*. Springer, 195–228.

[19] Hans-Martin Heyn, Eric Knauss, and Patrizio Pelliccione. 2023. A compositional approach to creating architecture frameworks with an application to distributed AI systems. *Journal of Systems and Software* 198 (2023), 111604.

[20] Ankur Jariwala, Aayushi Chaudhari, Chintan Bhatt, and Dac-Nhuong Le. 2022. Data Quality for AI Tool: Exploratory Data Analysis on IBM API. *International Journal of Intelligent Systems and Applications* 14, 1 (2022), 42.

[21] Mohammad Hossein Jarrahi, Ali Memariani, and Shion Guha. 2023. The Principles of Data-Centric AI. *Commun. ACM* 66, 8 (jul 2023), 84–92. https://doi.org/10.1145/3571724

[22] B. Kitchenham and S Charters. 2007. *Guidelines for performing systematic literature reviews in software engineering*. Technical Report. EBSE-2007-01.

[23] Dominik Kreuzberger, Niklas Kühl, and Sebastian Hirschl. 2023. Machine learning operations (mlops): Overview, definition, and architecture. *IEEE Access* (2023).

[24] Lucy Ellen Lwakatare, Ivica Crnkovic, and Jan Bosch. 2020. DevOps for AI– Challenges in Development of AI-enabled Applications. In *2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 1–6.

[25] Lucy Ellen Lwakatare, Ellinor Rånge, Ivica Crnkovic, and Jan Bosch. 2021. On the Experiences of Adopting Automated Data Validation in an Industrial Machine Learning Project. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. 248–257. https://doi.org/10.1109/ICSE-SEIP52600.2021.00034

[26] Juliette Mattioli, Gabriel Pedroza, Souhaiel Khalfaoui, and Bertrand Leroy. 2022. Combining Data-Driven and Knowledge-Based AI Paradigms for Engineering AI-Based Safety-Critical Systems. In *Workshop on Artificial Intelligence Safety (SafeAI)*.

[27] Marcel Meesters, Petra Heck, and Alexander Serebrenik. 2022. What is an AI engineer? An empirical analysis of job ads in The Netherlands. In *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*. 136–144.

[28] Aiswarya Raj Munappy, Jan Bosch, Helena Holmström Olsson, Anders Arpteg, and Björn Brinne. 2022. Data management for production quality deep learning models: Challenges and solutions. *Journal of Systems and Software* 191, 111359.

[29] Aiswarya Raj Munappy, David Issa Mattos, Jan Bosch, Helena Holmström Olsson, and Anas Dakkak. 2020. From ad-hoc data analytics to dataops. In *Proceedings of the International Conference on Software and System Processes*. 165–174.

[30] N. Nahar, H. Zhang, G. Lewis, S. Zhou, and C. Kastner. 2023. A Meta-Summary of Challenges in Building Products with ML Components – Collecting Experiences from 4758+ Practitioners. In *2023 IEEE/ACM 2nd International Conference on AI Engineering – Software Engineering for AI (CAIN)*. IEEE Computer Society, Los Alamitos, CA, USA, 171–183. https://doi.org/10.1109/CAIN58948.2023.00034

[31] Pia Niemelä, Bilhanan Silverajan, Mikko Nurminen, Jenni Hukkanen, and Hannu-Matti Järvinen. 2022. LAOps: Learning Analytics with Privacy-aware MLOps. In *International Conference on Computer Supported Education, CSEDU*. Science and Technology Publications (SciTePress), 213–220.

[32] Ipek Ozkaya. 2020. What is really different in engineering AI-enabled systems? *IEEE software* 37, 4 (2020), 3–6.

[33] Andrei Paleyes, Christian Cabrera, and Neil D Lawrence. 2022. An empirical evaluation of flow based programming in the machine learning deployment context. In *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*. 54–64.

[34] Patrick Petersen, Hanno Stage, Jacob Langner, Lennart Ries, Philipp Rigoll, Carl Philipp Hohl, and Eric Sax. 2022. Towards a Data Engineering Process in Data-Driven Systems Engineering. In *2022 IEEE International Symposium on Systems Engineering (ISSE)*. IEEE, 1–8.

[35] M Aiswarya Raj, Jan Bosch, Helena Holmström Olsson, and Anders Jansson. 2021. On the Impact of ML use cases on Industrial Data Pipelines. In *2021 28th Asia-Pacific Software Engineering Conference (APSEC)*. IEEE, 463–472.

[36] Joe Reis and Matt Housley. 2022. *Fundamentals of Data Engineering*. O'Reilly.

[37] Mehrnaz Sabet, Praveen Palanisamy, and Sakshi Mishra. 2023. Scalable modular synthetic data generation for advancing aerial autonomy. *Robotics and Autonomous Systems* 166 (2023), 104464.

[38] Johnny Saldaña. 2011. *The Coding Manual for Qualitative Researchers* (2nd editio ed.). SAGE Publications Inc. 329 pages.

[39] Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. 2021. "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.

[40] Sagar Sen, Erik Johannes Husom, Arda Goknil, Simeon Tverdal, Phu Nguyen, and Iker Mancisidor. 2022. Taming data quality in AI-enabled industrial internet of things. *IEEE Software* 39, 6 (2022), 35–42.

[41] Alex Serban, Koen van der Blom, Holger Hoos, and Joost Visser. 2020. Adoption and effects of software engineering best practices in machine learning. In *Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 1–12.

[42] Karthik Shivashankar and Antonio Martini. 2022. Maintainability Challenges in ML: A Systematic Literature Review. In *2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 60–67.

[43] Arumoy Shome, Luis Cruz, and Arie Van Deursen. 2022. Data smells in public datasets. In *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*. 205–216.

[44] Daniel Tebernum, Marcel Altendeitering, and Falk Howar. 2021. DERM: A Reference Model for Data Engineering.

[45] Saravanan Thirumuruganathan, Mayuresh Kunjir, Mourad Ouzzani, and Sanjay Chawla. 2021. Automated Annotations for AI Data and Model Transparency. *J. Data and Information Quality* 14, 1, Article 2 (dec 2021), 9 pages. https://doi.org/10.1145/3460000

[46] Yue Wang, Long Lin, He Yang, Cuncun Shi, and Weijiang Lu. 2022. The Construction Techniques of Artificial Intelligence Hierarchical Dataset in Power Industry. In *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Vol. 6. IEEE, 320–325.

[47] Stephen John Warnett and Uwe Zdun. 2022. Architectural design decisions for the machine learning workflow. *Computer* 55, 3 (2022), 40–51.

[48] Claes Wohlin. 2014. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. In *Proceedings of the International Conference on Evaluation and Assessment in Software Engineering (EASE)*. London (UK), Article 38, 10 pages.

[49] Haruki Yokoyama. 2019. Machine learning system architectural pattern for improving operational stability. In *2019 IEEE International Conference on Software Architecture Companion (ICSA-C)*. IEEE, 267–274.

[50] Jie M Zhang, Mark Harman, Lei Ma, and Yang Liu. 2020. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering* (2020).