Tools/frameworks that support development process of AI-based software: validations in white literature

Tugba Gurgen Erdogan[1,*], Haluk Altunel[2,3] and Ayca Kolukısa Tarhan[1]

[1] *Computer Engineering Department of Hacettepe University, Beytepe Campus, Ankara, 06800, Turkey*
[2] *Computer Engineering Department of Bilkent University, Bilkent, 06800, Turkey*
[3] *Softtech Inc Hacettepe Teknokent, 6.arge, Beytepe, Ankara, 06800, Turkey*

Abstract

Context: Artificial Intelligence (AI)-based software has gained increasing interest, especially in the last decade, due to advancements in underlying technologies and demands in varying business domains. With the proliferation to develop such software, there appears a need for developing methods and supporting tools/frameworks. Purpose: In this paper, we focus on tools/frameworks to automate AI-based software development process, from a holistic view. We review the scientific studies that were empirically validated and also evaluate their proposals with respect to basic characteristics including theme, research methods, types, domains, and a number of cases in empirical validations. Method: We elicit relevant studies (with the contribution type of tool or framework) from a larger set of primary studies identified by a systematic literature review on AI-based software development process. We select 14 primary studies in this context and analyze them with respect to the purposes of the proposals. Results: We review tools/frameworks that support AI-based software development process under four headings: software system development process, the development process of fair software, model development process, and model deployment and operation processes. We observe that domains of empirical validation are diverse while the number of empirical cases applied for validation is limited. Also, only half of the primary studies provide links to their proposals as open-source, which is very important for the repeatability of the empirical validations.

Keywords

Tool, framework, artificial intelligence, machine learning, software development, development process

1. Introduction

Almost every day, we hear about intelligent systems such as self-driving cars and unmanned aerial vehicles that come to life with artificial intelligence. In addition and in parallel to the era of digitalization, the demand for smart systems that analyze existing data and turn it into an advantage for institutions and customers in many business areas such as trade, health, finance, production, education, etc. is increasing. Accordingly, the growing scale of Internet-based systems and innovations in social platforms, autonomous systems and cloud computing, and

IWSM-MENSURA'22: The Joint Conference of the 31th International Workshop on Software Measurement (IWSM) and the 16th International Conference on Software Process and Product Measurement (MENSURA) Sep 28–30, 2022, Izmir, Turkey

[*] Corresponding author.

tugba@cs.hacettepe.edu.tr (T. G. Erdogan); altunel@bilkent.edu.tr (H. Altunel); atarhan@cs.hacettepe.edu.tr (A. K. Tarhan)
0000-0003-1491-8739 (T. G. Erdogan); 0000-0003-1103-3644 (H. Altunel); 0000-0003-1466-9605 (A. K. Tarhan)

CEUR Workshop Proceedings (CEUR-WS.org)

the advancement of artificial intelligence (AI) and machine learning (ML) techniques in the last decade, have created a new specialty for the software industry: AI-based software system development.

Several studies have reported challenges about AI-based software development, more frequently for ML software [1, 2, 3] or deep learning (DL) software [4], and more specifically for development phases such as requirements [5] or testing [6]. In addition, large companies such as Google and Microsoft have shared their experiences of AI-based (or data-centric) software development in recent years. For example, Google reported that machine learning software used in everyday life will likely require ongoing maintenance costs [7]. Microsoft reported about the challenges AI teams face in managing data, component abstraction, and reusing software, noting that customizing and reusing AI-based software requires different methods and skills than those applied in traditional software development [8]. Consequently, development methods and supporting technologies must be adapted to meet the specific challenges of AI-based software engineering.

As a response to the need stated above, in this paper, we focus on and review in detail the studies in scientific (or white) literature, which propose the tools/frameworks to support the development process of AI-based software. We select 14 studies that have been empirically validated from a larger set of primary studies identified by a systematic literature review (SLR) on AI-based software development process. We review the studies by grouping them according to their purposes of proposals and evaluate their proposals with respect to basic characteristics including theme, research methods and types, domains and number of cases in empirical validations, contexts of empirical validations, and open links to the proposed tools/frameworks.

The rest of this paper is organized as follows. Section 2 provides a summary of the related studies. Section 3 overviews the study search and selection processes, and potential validity threats. Section 4 reviews the primary studies in detail, and Section 5 evaluates them with respect to basic characteristics of validation. Finally, Section 6 concludes the paper.

2. Related Work

A list of related secondary studies, which include primary studies on tools/frameworks for AI-based software development, are provided in Table 1. As seen from the table, there is no study that solely investigates, reviews in detail, and evaluates the tools/frameworks which support the AI-based software development process. Therefore, ours is the first study that is intended for this specific purpose.

3. Methodology

First, we have applied guidelines to conduct systematic literature reviews in software engineering [17], and have searched and identified 85 primary studies that propose development models or processes for AI-based software. In search and study selection, we have not focused on specific phases of software development such as requirements or design; rather, we took a holistic view for the software development process.

Table 1
Related secondary studies

Ref.	Focus
[9]	This survey study presents a comprehensive overview of recent and cutting-edge AI software, including open-source frameworks and libraries, which support the implementation of ML and DL algorithms. However, it does not specifically focus on the development process of these algorithms.
[10]	This study mines both academic and grey literature, identifies 29 engineering best practices for ML applications, and conducts a survey among 300+ practitioners to determine the degree of adoption of these practices. Although it refers to a number of tools in both white and grey literature, its focus is not on tools/frameworks that support the development process, and it does not include a detailed review.
[11]	This survey describes major research efforts where ML systems have been deployed at the edge of computer networks, focusing on the operational aspects including compression techniques, tools, frameworks, and hardware used in successful applications of intelligent edge systems. Yet, it does not focus on tools/frameworks to support the development or deployment processes.
[12]	This study summarizes secondary data from non-grey and grey academic literature. It identifies, analyzes, and synthesizes the challenges of ML-enabled software development, and re-frames the traditional SE development process to engineer the ML software development process. However, it does not specifically review the primary studies that propose tools/frameworks to support the development process.
[13]	This study presents the results from a systematic mapping study on the life cycle of AI models. It yields 405 primary studies, mapped in 5 different research topics. It reports that only a minority of publications focus on data management and model production problems, and that more studies should address the AI life cycle from a holistic perspective. This study does not have a specific focus on tools/frameworks that support the life cycle of AI models.
[3]	This study identifies and synthesizes the current state of software engineering (SE) research for engineering ML systems by a systematic literature review of 141 primary studies. It highlights that SE aspects do not have a mature set of tools and techniques. It reviews in detail the primary studies under the knowledge areas of SWEBOK [14], including the ones that propose tools/frameworks in each area. However, only few of these primary studies support the development process of ML software.
[15]	The study surveys existing efforts for DevOps, and categorizes them according to the life cycle stages that they contribute. It outlines directions for future work in quality-aware DevOps, in particular at AI for DevOps and DevOps for AI software. In terms of tools/frameworks, it refers to only one primary study which proposes a software architecture solution that can ensure the continuous development of computer vision applications.
[16]	To synthesize knowledge on SE approaches for building, operating, and maintaining AI-based systems, this study presents results from a systematic mapping of 248 primary studies. It classifies their subjects to SWEBOK's knowledge areas, including the "Software Engineering Process" area and its supporting tools/frameworks. However, it only refers to these studies in the classification scheme and does not include a detailed review.

3.1. Research Questions

Research questions (RQs) of this review study are described below:

RQ1. What is the type of the proposed tool/framework? (plugin, library, toolkit, etc.)

RQ2. For what purpose, theme (data, model, maturity, etc.) or domain (e.g., health) has the tool/framework been proposed?

RQ3. Does the proposal include a tool? If yes, is the tool accessed publicly or is it open-source? If yes, what is its url?

RQ4. Has the proposal been validated (with weak or strong empirical study)? If yes, what are the research methods (case study, survey etc.) used in and the context (scope, etc.) of validation?

3.2. Search strategy and study selection

Search sentences that were run in digital libraries as of January 2022 are listed below:

1. ("artificial intelligence software" OR "AI software" OR "machine learning software" OR "ML software") AND ("development approach" OR "development process" OR "development model" OR "development life-cycle")
2. ("artificial intelligence applications" OR "AI applications" OR "machine learning applications" OR "ML applications") AND ("development approach" OR "development process" OR "development model" OR "development life-cycle")
3. ("artificial intelligence software" OR "AI software" OR "machine learning software" OR "ML software") AND ("software development" OR "application development") AND ("approach" OR "process" OR "model" OR "life-cycle")
4. ("artificial intelligence applications" OR "AI applications" OR "machine learning applications" OR "ML applications") AND ("software development" OR "application development") AND (approach OR process OR model OR life-cycle)
5. ("artificial intelligence software" OR "AI software" OR "machine learning software" OR "ML software") AND (maturity OR capability) AND ("software development" OR "software engineering") AND (approach OR process OR model OR life-cycle)
6. ("artificial intelligence applications" OR "AI applications" OR "machine learning applications" OR "ML applications") AND (maturity OR capability) AND ("software development" OR "software engineering") AND (approach OR process OR model OR life-cycle)
7. ("artificial intelligence software" OR "AI software" OR "machine learning software" OR "ML software") AND (challenges OR difficulties OR problems) AND ("software development" OR "software engineering")
8. ("artificial intelligence applications" OR "AI applications" OR "machine learning applications" OR "ML applications") AND (challenges OR difficulties OR problems) AND ("software development" OR "software engineering")

The number of studies retrieved from the digital libraries by these search sentences are given in Table 2. While identifying and selecting the primary studies, we have eliminated studies applying inclusion and exclusion criteria in the order of the digital libraries shown in the table. The inclusion criteria are: I1) papers focus on the holistic view of software development process rather than specific phases of software development; I2) papers are in the area of both software engineering and AI. The exclusion criteria are: E1) publication language being other than English; E2) full-text not being available; E3) books and theses; E4) duplicate results from different search methods, and E5) papers are explicitly short (e.g., less than six pages).

Until now, we have identified contributions made and research methods [18] employed in the primary studies, which are listed in Table 3. Full details can be found in the study's transparent online spreadsheet (https://tinyurl.com/ytnxhhuy). Here, we should note that we are still in the data extraction stage for the larger pool, and present the initial results from it on the basis of supporting tools/frameworks in this paper.

Table 2
Number of primary studies retrieved and selected

Sentence	Number of initially retrieved studies						Number of selected studies for AI-based software development process	Number of selected primary studies to review in this paper
	Web of Science	Scopus	IEEE Xplore	Springer Link	ACM DL	Google Scholar		
S1	5	116	706	196	128	3690	17	2
S2	26	52	302	141	259	7760	20	3
S3	22	33	237	345	215	6190	12	2
S4	33	85	27	915	408	10100	4	1
S5	5	8	44	395	232	3930	13	5
S6	4	9	8	1343	428	7540	4	0
S7	66	43	762	997	357	8370	10	1
S8	34	81	90	4065	609	14800	5	0
Total	**15**	**8**	**15**	**4**	**38**	**5**	**85**	**14**

Table 3
Types of contributions and research methods used in classifying primary studies

Types of contributions	**Types of research**	**No.of studies per research method in this paper**
Tool/Framework Tasks/Phases Workflow Taxonomy Guideline List of challenges Solutions-to challenges Language Lessons learned List of best practices Empirical study only	(1) Solution proposal, (2) Validation research, (3) Evaluation research, (4) Experience paper.	(1) 5 (excluded in this paper), (2) 6, (3) 8, (4) 0.

In order to construct the study pool in line with the purpose of this paper, we have selected 14 primary studies that have a contribution type of tool or framework and a research method type of validation or evaluation. In validation research, techniques investigated are novel and have not yet been implemented in practice (used for example experiments, i.e. work done in the lab) while in evaluation research, techniques are implemented in practice (i.e. solution implementation) and an evaluation of the technique is conducted in terms of benefits and drawbacks (i.e. implementation evaluation) [19]. The reason for selecting a subset of primary studies with the stated characteristics is that, as we already mentioned in the introduction section, we would like to identify, review and evaluate the empirically validated tools/frameworks that support the development process of AI-based software.

3.3. Potential threats to validity

A number of validity threats are of concern for this study as adopted from [20, 21] in terms of construct, internal, external and conclusion validity. The guideline for conducting SLRs has been followed and a research protocol has been developed and reviewed to cope with construct validity. One threat of this type regarding the applied protocol can be excluding the studies for the specific phases of software development, such as requirements and design. However, we have done this purposefully, since the subject area is quite broad and we have considered the software development process as a whole. With this strategy applied, we have not only retrieved the studies that propose development models or processes as a primary contribution but also the studies that embed such contributions secondarily into their AI-based software development efforts. Another threat of construct validity is related to the research method (i.e. SLR) that we have employed in this study. Including the grey literature in addition to the white literature could have elicited further proposals of tools/frameworks for the AI-based software development process. We claim this threat so that it can be considered while reading our findings and also be addressed in future studies. In terms of internal validity, we have applied independent voting among authors for inclusion/exclusion of the studies and for quality assessment of initially included studies. Additionally, we have carried out pilot data extraction and have conducted regular meetings to ensure consistency and commonality during data extraction. Regarding external validity, we admit that the results we obtained are valid only for the primary study pool that we have created, and it cannot be generalized in other contexts. Finally, in terms of conclusion validity, we have made several meetings to analyze and synthesize the results of data extraction. Since we have used a standardized format (i.e. Google Sheet) in data analysis, we believe other researchers with the same study pool would come up with similar results.

4. Review of Tools/Frameworks

In this section, we review the selected primary studies on tools/frameworks that support AI-based software development process by grouping the studies under four headings. We performed thematic synthesis [22] to group the studies as: software system development process, development process of fair software, model development process, and model deployment and operation processes. The studies falling under these headings are elaborated in the following subsections.

Full details with respect to the research questions raised can be found in the study's online spreadsheet (https://tinyurl.com/ytnxhhuy) in the following columns, respectively: RQ1: Type (AD); RQ2: Purpose (AO), theme (AR), domain (AV); RQ3: Link to download (AQ); RQ4: Validation type (Z, AA), research methods (AB-AK), context (AV).

4.1. Tools/Frameworks That Support Software System Development Process

The primary studies grouped under this heading are summarized in Table 4, each with its title, purpose and scope. As seen from the table, these three studies have different concerns, which indicates that there is a diversity of the studies in this field. The first one is related to health informatics, and we know there is a trend in applying AI techniques in this domain to reveal important insights about healthcare data and healthcare processes. The second one is about deep learning-based project development using the principles of MDE (Model Driven Engineering) to solve the challenges to build the deep learning-based project development, particularly for neural networks which have complicated architecture and dependencies. The third one is seeking solutions for the scalability and performance problems for the development of conversational AI software like mobile software, and proposes a framework inspiring Cell theory and BDI (Belief-Desire-Intention) software model. These proposals can be considered as example solutions to common problems in AI software development, and as the trending topics for the researchers.

Table 4
Studies that support software system development process

Study, Year	**Purpose and Scope**
P1[23], 2020	- It investigates the interaction between software engineering and machine learning within the context of health systems. - It proposes SEMLHI framework (the framework and methodology of software engineering for ML in health informatics) that includes four modules: software, machine learning, machine learning algorithms, and health informatics data. - The SEMLHI methodology includes seven phases: designing, implementing, maintaining and defining workflows; structuring information; ensuring security and privacy; performance testing and evaluation; and releasing the software applications.
P4[24], 2021	- It supports deep learning based project development. - It proposes an ML-oriented artifact model inspiring Model Driven Development (MDE) and the evaluation of the proposed concepts are combined into a Maven based build infrastructure. - A Maven plugin is developed to automate build which provides install and deploy goals for all kinds of archives of the proposed artifact model.
P5[25], 2021	- It presents the CellS framework to improve smart software development on multicore mobile processor systems. - CellS framework has five important components: anima, delibera, cell, plan and ligand. - It is used in a resources-constrained mobile system for conversational AI software like a software dialogue system.

4.2. Tools/Frameworks That Support Development Process of Fair Software

The primary studies grouped under this heading are summarized in Table 5. It is apparent from the table that most of the studies focus on bias mitigation in ML software. Measurement of bias is handled in a maturity framework from the organizational level by applying ethical guidelines and standards, and fairness measurement is handled through a scalable measurement system for computing fairness metrics. It is obvious that ethical AI activities should be addressed with bias detection, bias/fair measurement, and bias mitigation methods starting from the initial stages of the AI development life cycle.

Table 5
Studies that support development process of fair software

Study, Year	Purpose and Scope
P3[26], 2019	- It presents a maturity framework based on ethical principles and best practices to evaluate the organization capability in order to effectively govern AI bias. - It provides a measurement of bias governance capability maturity of the organizations by separating the scoring in two sections as "consideration" and "action".
P8[27], 2021	- It introduces Fairea, a model behaviour mutation approach for benchmarking and quantitatively evaluating bias mitigation methods for ML software. - It includes three bias mitigation approaches: pre-processing, in-processing and post-processing. - There are three primary steps: baseline creation with model behaviour mutation, bias mitigation effectiveness region division, and quantitative evaluation of trade-off effectiveness.
P9[28], 2020	- It proposes a method named Fairway for bias detection and bias mitigation in binary classification models to remove ethical bias from training data and trained models. - This handy tool combines two bias mitigation approaches: pre-processing (before model training) and in-processing (while model training).
P11 [29], 2020	- It presents LinkedIn Fairness Toolkit (LiFT), a framework for scalable computation of fairness metrics as part of ML systems. - It highlights the key requirements in deployed settings, and presents the design of a fairness measurement system. - LiFT comprises bias measurement and mitigation components that can be integrated into different stages of an ML training and serving system.

4.3. Tools/Frameworks To Support Model Development Process

The primary studies grouped under this heading are summarized in Table 6. As seen from the table, even though introduced models have different specific targets, they all have the common focus on data contribution in model development. The first model focuses on the data scientists' cognitive workflows that include data collection. The second model introduces a validation framework for data. The last model provides an open source library with different data modalities.

Table 6
Studies that support model development process

Study	Purpose and Scope
P6[30], 2021	- It focuses on observing and analyzing data scientists' cognitive workflows as they develop predictive models. - It proposes DSWorkFlow that covers data collection,workflow reconstruction, and feature extraction stages.
P12[31], 2021	- It helps in systematizing the adoption of data validation processes and tools in industrial ML projects. - It introduces a data validation framework (DVF) that includes validation process, validation artifacts, data validation types, data validation tool setup, and feedback and mitigation strategy.
P13[32], 2020	- It provides guidance to develop, and metrics to evaluate. - It comes up with Machine Learning (ML) Bazaar as open source libraries for components to create general-purpose, multi-task, end-to-end AutoML systems that provide solutions to different data modalities (image, text, graph, tabular, relational, etc.) and problem types (classification, regression, anomaly detection, graph matching, etc.).

4.4. Tools/Frameworks That Support Model Deployment and Operation Processes

The primary studies grouped under this heading are summarized Table 7. It is apparent from the table that all studies for model deployment and operation take software, data and its machine learning models into account together. That means pipeline and deployment of software are not independent of data and machine learning models. In addition to that, the use cases are limited and only selected models are used. They need to be applied in different domains with real use cases to get wider insight for further usage areas.

5. Evaluation of Tools/Frameworks

After grouping and reviewing the primary studies in the previous section, we further evaluate their proposals with respect to basic characteristics of empirical validations, which include research method and type, domain and number of empirical cases, context of empirical cases, and link to proposed tool/framework. Table 8 presents these characteristics for all the included primary studies in columns.

As seen from the Table 8, the evaluation type of research (8) is slightly more than validation type (6) in the primary studies. Case study research (in eight studies) and lab experiments (in six studies) are the most frequently used research methods. While domains of empirical validation are diverse, the number of empirical cases applied for validation is limited. Contexts of empirical validation are various in alignment with the diversity of the domains. Finally, we see that only half (7) of the primary studies provide links to their proposals as open source, which is very important for the repeatability of the empirical validations as well as for further validation studies by other researchers or practitioners.

Table 7
Studies that support model deployment-operation processes

Study, Year	Purpose and Scope
P2[33], 2020	- It proposes a framework for selecting a certain architecture. - It outlines the trade-off between device cost, model performance, and data privacy in selection of architecture alternatives. - Total of five different architectural alternatives exist from centralized cloud to fully decentralized edge architectures. It explains seven case companies in the embedded system domain.
P7[34], 2021	- It presents a framework for supporting collaborative data science development and open-source feature engineering. - The framework's name is Ballet that includes feature definition, feature engineering pipeline, feature API validation, ML performance validation, and project management capabilities. - It presents two use cases, first one is in disease prediction and the other one is in income prediction.
P10[35], 2020	- It introduces a development environment namely Gestalt that allows developers to implement a classification pipeline, analyzes data as it moves through that pipeline, and supports easy transition between implementation and experiment. - It explains 2 use cases: sentiment analysis, gesture recognition. - It underlines the contributions of flexibility and visualization.
P14[36], 2019	- It introduces a cognitive hardware and software composed infrastructure. - The infrastructure is CHASE-CI for managing fast GPU appliances for machine learning and storage managed through Kubernetes on the high-speed. - It presents a software containerization approach and libraries for turning into a distributed computer for big data analysis. - It explains a use case in earth science phenomena with object segmentation workflow.

In terms of the quantity of the cases included in empirical validation, the following studies come to the fore, only the first one employing validation type research: P2 [33] (for model deployment and operation processes) with seven cases from software intensive embedded system domain, P9 [28] (for development process of fair software) with five cases regarding five datasets used from UC Irvine ML Repository, and P13 [32] (for model development process) with five cases from mixed domains including telemetry, healthcare, energy, water supply, and data science.

6. Conclusion

In this study, we reviewed tools/frameworks within white literature in a systematic way for tools/frameworks that support the AI-based software development process from a holistic perspective. We selected and analyzed 14 primary studies and we grouped them under four headings: software system development process, the development process of fair software, model development process, and model deployment and operation processes.

Within these four groups, the scopes of the studies are seen as diversified, and data and model dependency of the tools or frameworks is observed such as P1 proposing a framework for health informatics whereas P5 is focusing on mobile systems. Additionally, data and machine learning models have effects on the software development process, and only half of the studies include the links to their proposals. On the other hand, more than half of the studies are presented with evaluations while the rest of them use validation type research.

These findings underline the fact that the AI-based software development process is an emerging field of research and needs further investigation. Furthermore, some of the proposed tools or frameworks need to be sharpened and to be validated in different domains in various industrial contexts. For researchers and practitioners, the next step can be the comparison of new tools/frameworks with existing models in industry in terms of performance and quality metrics. Additionally, a comparison of the tools/frameworks that support traditional software development processes may provide valuable findings in the maturation of the overall software development in the AI/ML area. As the last point, practitioners in industry can test the seven of the tools/frameworks with their access links given in Table 8, evaluate their real-life performances, and share their results publicly as the feedback to the community as well as the researchers who introduced them.

References

[1] L. E. Lwakatare, A. Raj, J. Bosch, H. H. Olsson, I. Crnkovic, A taxonomy of software engineering challenges for machine learning systems: An empirical investigation, in: International Conference on Agile Software Development, Springer, Cham, 2019, pp. 227–243.

[2] F. Kumeno, Sofware engneering challenges for machine learning applications: A literature review, Intelligent Decision Technologies 13 (2019) 463–476.

[3] G. Giray, A software engineering perspective on engineering machine learning systems: State of the art and challenges, Journal of Systems and Software 180 (2021) 111031.

[4] A. Arpteg, B. Brinne, L. Crnkovic-Friis, J. Bosch, Software engineering challenges of deep learning, in: 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), IEEE, 2018, pp. 50–59.

[5] H. Belani, M. Vukovic, Ž. Car, Requirements engineering challenges in building ai-based complex systems, in: 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW), IEEE, 2019, pp. 252–255.

[6] K. Sugali, Software testing: Issues and challenges of artificial intelligence & machine learning, IJAIA (2021).

[7] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, D. Dennison, Hidden technical debt in machine learning systems, Advances in neural information processing systems 28 (2015).

[8] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, T. Zimmermann, 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, 2019, pp. 291–300.

[9] G. Nguyen, S. Dlugolinsky, M. Bobák, V. Tran, Á. López García, I. Heredia, P. Malík, L. Hluchỳ, Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey, Artificial Intelligence Review 52 (2019) 77–124.

[10] A. Serban, K. van der Blom, H. Hoos, J. Visser, 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), 2020, pp. 1–12.

[11] M. S. Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, F. Hussain, Machine learning at the network edge: A survey, ACM Computing Surveys (CSUR) 54 (2021) 1–37.

[12] S. Saeed, M. M. Abubakar, M. Karabatak, Software engineering for data mining (ml-enabled) software applications, in: 2021 9th International Symposium on Digital Forensics and Security (ISDFS), IEEE, 2021, pp. 1–9.

[13] Y. Xie, L. Cruz, P. Heck, J. S. Rellermeyer, Systematic mapping study on the machine learning lifecycle, in: 2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN), IEEE, 2021, pp. 70–73.

[14] A. Abran, et al., Swebok, Guide to the Software Engineering Body of Knowledge (2004).

[15] A. Alnafessah, A. U. Gias, R. Wang, L. Zhu, G. Casale, A. Filieri, Quality-aware devops research: Where do we stand?, IEEE Access 9 (2021) 44476–44489.

[16] S. Martínez-Fernández, J. Bogner, X. Franch, M. Oriol, J. Siebert, A. Trendowicz, A. M. Vollmer, S. Wagner, Software engineering for ai-based systems: a survey, ACM Transactions on Software Engineering and Methodology (TOSEM) 31 (2022) 1–59.

[17] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering (2007).

[18] R. Wieringa, N. Maiden, N. Mead, C. Rolland, Requirements engineering paper classification and evaluation criteria: a proposal and a discussion, Requirements engineering 11 (2006) 102–107.

[19] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, in: 12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12, 2008, pp. 1–10.

[20] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in software engineering, Springer Science & Business Media, 2012.

[21] X. Zhou, Y. Jin, H. Zhang, S. Li, X. Huang, A map of threats to validity of systematic literature reviews in software engineering, in: 2016 23rd Asia-Pacific Software Engineering Conference (APSEC), IEEE, 2016, pp. 153–160.

[22] D. S. Cruzes, T. Dyba, Recommended steps for thematic synthesis in software engineering, in: 2011 international symposium on empirical software engineering and measurement, IEEE, 2011, pp. 275–284.

[23] M. Moreb, T. A. Mohammed, O. Bayat, A novel software engineering approach toward using machine learning for improving the efficiency of health systems, IEEE Access 8 (2020) 23169–23178.

[24] A. Atouani, J. C. Kirchhof, E. Kusmenko, B. Rumpe, Artifact and reference models for generative machine learning frameworks and build systems, in: Proceedings of the 20th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences, 2021, pp. 55–68.

[25] C.-H. Chen, M.-C. Wu, Cells: A cell-inspired efficient software framework for ai-enabled

application on resources-constrained mobile system, Electronics 10 (2021) 568.

[26] D. L. Coates, A. Martin, An instrument to evaluate the maturity of bias governance capability in artificial intelligence projects, IBM Journal of Research and Development 63 (2019) 7–1.

[27] M. Hort, J. M. Zhang, F. Sarro, M. Harman, Fairea: A model behaviour mutation approach to benchmarking bias mitigation methods, in: Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2021, pp. 994–1006.

[28] J. Chakraborty, S. Majumder, Z. Yu, T. Menzies, Fairway: A way to build fair ml software, in: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2020, pp. 654–665.

[29] S. Vasudevan, K. Kenthapadi, Lift: A scalable framework for measuring fairness in ml applications, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 2773–2780.

[30] M. Mash, S. Rosenthal, R. Simmons, Dsworkflow: A framework for capturing data scientists' workflows, in: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–7.

[31] L. E. Lwakatare, E. Rånge, I. Crnkovic, J. Bosch, 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), IEEE, 2021, pp. 248–257.

[32] M. J. Smith, C. Sala, J. M. Kanter, K. Veeramachaneni, The machine learning bazaar: Harnessing the ml ecosystem for effective system development, in: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, 2020, pp. 785–800.

[33] M. M. John, H. H. Olsson, J. Bosch, Ai deployment architecture: Multi-case study for key factor identification, in: 2020 27th Asia-Pacific Software Engineering Conference (APSEC), IEEE, 2020, pp. 395–404.

[34] M. J. Smith, J. Cito, K. Lu, K. Veeramachaneni, Enabling collaborative data science development with the ballet framework, Proceedings of the ACM on Human-Computer Interaction 5 (2021) 1–39.

[35] K. Patel, N. Bancroft, S. M. Drucker, J. Fogarty, A. J. Ko, J. Landay, Gestalt: integrated support for implementation and analysis in machine learning, in: Proceedings of the 23nd annual ACM symposium on User interface software and technology, 2010, pp. 37–46.

[36] I. Altintas, K. Marcus, I. Nealey, S. L. Sellars, J. Graham, D. Mishin, J. Polizzi, D. Crawl, T. DeFanti, L. Smarr, Workflow-driven distributed machine learning in chase-ci: A cognitive hardware and software ecosystem community infrastructure, in: 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), IEEE, 2019, pp. 865–873.

Table 8
Evaluation of the primary studies with respect to basic characteristics of empirical validation

Study	Related Section and Theme	Related Research Method Type	Research Methods applied for empirical validation	Domains of empirical validation (and the number of empirical cases)	Context of empirical validation	Link(s) to access
P1 [23]	IV.A-Sw.Sys. Dev.	Validation	Case Study	Healthcare (1)	Three hospitals and nine medical centers were used as the basis of the dataset.	Not available (N/A)
P2 [33]	IV.D-Model Dep.Op.	Validation	Case Study, Interview, Workshop	Software Intensive Embedded System (7)	1) mobility solutions for vehicle development 2) services, software and infrastructure in communication technology 3) pumbs and electronics for pumb control 4) services in healthcare and energy 5) packaging and processing solution for food products 6) manufacturing and marketing vehicles 7) systems for power generation, transmission and medical diagnosis	N/A
P3 [26]	IV.B-Dev.of Fair Sw.	Evaluation	Case Study, Interview, Ques.Survey	Data Science (3)	Three independent AI software projects were identified to demonstrate variances between organizational size, industry, project purpose and project stage.	N/A
P4 [24]	IV.A-Sw.Sys. Dev.	Validation	Case Study	Data Science (3)	1) Evolution example (calculator that can work with handwritten input) 2) Neural network composition (sentiment of textual input) 3) Reference model-based training configuration (autonomous driving domain)	N/A
P5[25]	IV.A-Sw.Sys. Dev.	Evaluation	Lab Experiment	Software System (2)	1) hypothetical dialogue system 2) AI-enabled dialogue application on resources-constrained mobile system	N/A
P6 [30]	IV.C-Model Dev.	Validation	Lab Experiment	Data Science (7)	Seven data scientists, each created three machine learning models	https://github.com/ MoshikMash/DSWorkFlow
P7 [34]	IV.D-Model Dep.Op.	Validation	Case Study, Lab Experiment	Healthcare (1), Real-estate (1), Census-income (1).	1) lab exp. 1: disease incident prediction 2) lab exp. 2: house price prediction 3) case study: predict- census-income	https://github.com/ballet
P8 [27]	IV.B-Dev.of Fair Sw.	Evaluation	Lab Experiment	Census-income (1), Criminal (1), Finance (1).	Three datasets in fairness literature: 1) Adult census income 2) COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) 3) German credit data	https://github.com/ maxhort/Fairea/
P9 [28]	IV.B-Dev.of Fair Sw.	Evaluation	Lab Experiment	Census-income (1), Criminal (1), Finance (2), Healthcare (1).	Five datasets from UCI ML Repository: 1) Adult census income 2) COMPAS 3) German credit data 4) Default credit 5) Heart health	https://github.com/ joymallyac/Fairway
P10 [35]	IV.D-Model Dep.Op.	Validation	Lab Experiment	Data Science (2)	1) sentiment analysis 2) gesture recognition	N/A
P11 [29]	IV.B-Dev.of Fair Sw.	Evaluation	Real-World Experiment	Data Science (4)	1-3) Three web-scale ML pipelines at LinkedIn 4) Adult census income dataset from UC Irvine ML Repository	https://github.com/ Linkedin/LiFT
P12 [31]	IV.C-Model Dev.	Evaluation	Action Research, Case Study	Telecommunication (1)	Classifying faults from returned hardware telecommunication devices, in a large software-intensive organization.	N/A
P13 [32]	IV.C-Model Dev.	Evaluation	Case Study, Real-World Experiment	Telemetry (1), Healthcare (1), Energy (1), Water Supply (1) Data Science (1).	1) anomaly detection for satellite telemetry 2) predicting clinical outcomes from electronic health records 3) failure prediction in wind turbines 4) leaks and crack detection in water distribution systems 5) DARPA D3M Program	https://github.com/HDI-Project/MLBlocks https://github.com/HDI-Project/BTB https://github.com/HDI-Project/AutoBazaar
P14 [36]	IV.D-Model Dep.Op.	Validation	Case Study	Atmospheric Science (1)	Object segmentation workflow (calculating Integrated Water Vapor Transport from the assimilated meteorological field data archive)	http://ucsd-prp.gitlab.io/ nautilus/namespaces/