Intelligent Decision Technologies 13 (2019) 463–476
DOI 10.3233/IDT-190160
IOS Press

Software engineering challenges for machine learning applications: A literature review

Fumihiro Kumeno
Department of Information Technology and Media Design, Nippon Institute of Technology, Miyashiro-machi, Minamisaitama-gun, Saitama Pref, 345-8501, Japan
E-mail: kumeno@nit.ac.jp

Abstract. Machine learning techniques, especially deep learning, have achieved remarkable breakthroughs over the past decade. At present, machine learning applications are deployed in many fields. However, the outcomes of software engineering researches are not always easily utilized in the development and deployment of machine learning applications. The main reason for this difficulty is the many differences between machine learning applications and traditional information systems. Machine learning techniques are evolving rapidly, but face inherent technical and non-technical challenges that complicate their lifecycle activities. This review paper attempts to clarify the software engineering challenges for machine learning applications that either exist or potentially exist by conducting a systematic literature collection and by mapping the identified challenge topics to knowledge areas defined by the Software Engineering Body of Knowledge (Swebok).

Keywords: Machine learning, software engineering challenges, Swebok, systematic literature review

1. Introduction

Software systems with intelligent components based on machine learning (ML) techniques have been widely developed and are now applied in various fields, such as electronic commerce, finance, manufacturing, healthcare, entertainment, and the automotive industry. These practical applications (ML applications) have been anchored by significant advances in ML techniques and software platforms for ML development. ML techniques have been copiously researched and published over a broad range of topics. In particular, the breakthrough in deep learning research is the driving force behind the advance of ML techniques. Many papers on deep learning techniques, including learning algorithms, performance improvement, evaluations, and applications, have been extensively published.

However, the systematic development, deployment and operation of ML applications faces major difficulties (e.g., [1–3,18,21]). The methodologies and tools of software engineering (SE) have greatly contributed to a wide range of activities in the lifecycles of traditional information systems, but are difficult to implement in ML application projects because ML applications and traditional software systems differ in fundamental ways. An ML application involves at least a computational model (an ML model) which is trained on some training data, and which processes additional data to make some inferences. The behavior of an ML model-based program depends on the training data, and is often unpredictable. This phenomenon introduces various uncertainties into the system's outcomes [4,5]. The lifecycle process of ML applications also differs from that of traditional software processes. Figure 1 is a simplified workflow diagram of a supervised ML application. The workflow comprises a requirements analysis, data-oriented works, model-oriented works, and DevOps works. The requirements analysis performs the data-analysis activities of the system requirements and the following data-oriented works. The data-oriented works include the data collection, data validation, data cleaning, and feature extraction. Model-oriented works cover the model design and construction, model training, evaluation, and optimization. Finally, the DevOps works cover activities such as model deployment, monitoring, control, and retraining. The workflow includes many feedback loops. Note that the model evaluation and monitoring may loop back to any of the previous works, and the model training may loop back to feature extraction.

Machine learning algorithms, models and related techniques are rapidly evolving and new challenges are emerging. Such situations make software engineering practices for ML applications more difficult activities.

ISSN 1872-4981/$35.00

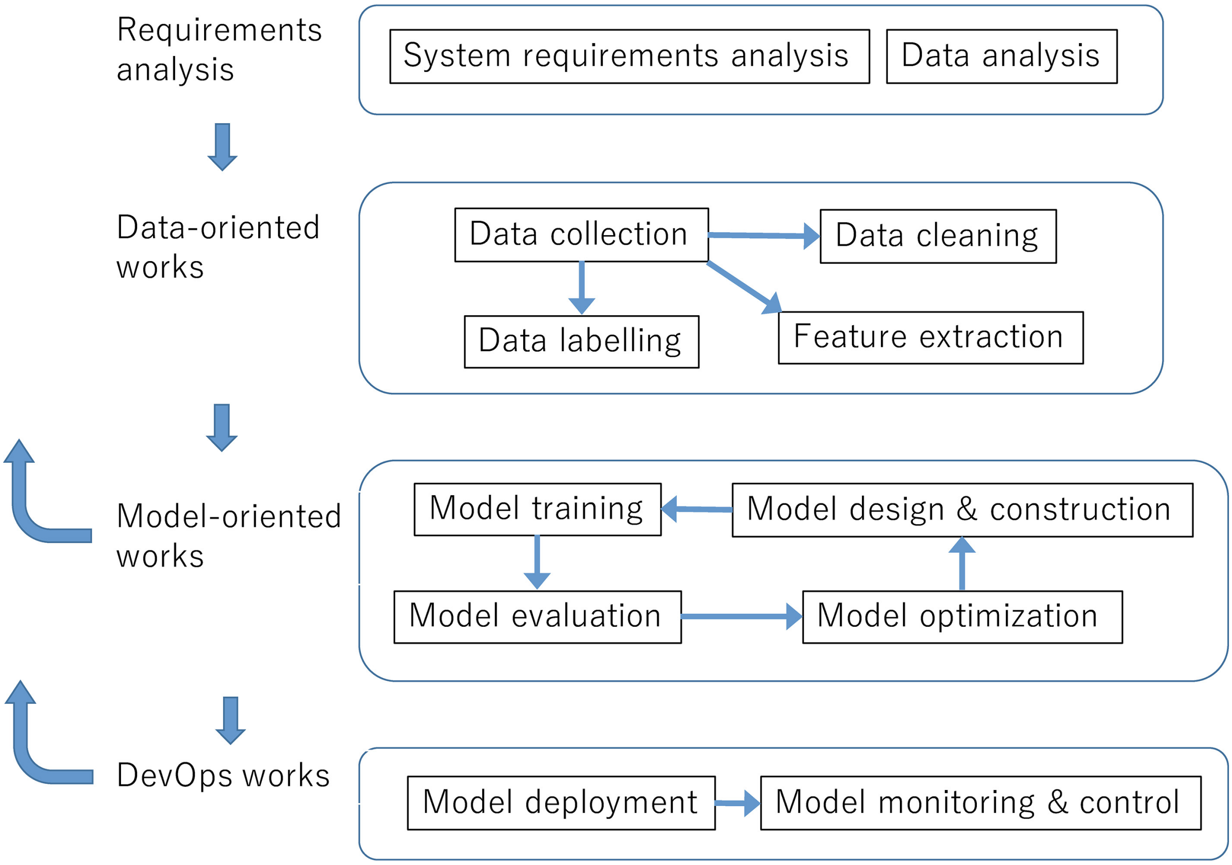


Fig. 1. A workflow example of supervised machine learning applications.

Given the various challenges in software engineering of ML, we surmise that SE challenges for ML applications cover a similarly wide range of topics. SE challenges for ML applications have been discussed in many papers [15–18,21], but to our knowledge, no survey paper has clarified the overview of SE challenges for ML applications, that is, what SE challenges have been discussed? and which SE research topics are closely related to each challenge?

The Software Engineering Body of Knowledge (Swebok) [6] classifies software engineering topics into knowledge areas. We presume that this comprehensive framework is helpful to seek answers to the following research questions.

- RQ1: What SE challenges for ML applications have been discussed and potentially exist?
- RQ2: Which knowledge area is closely related to each of them?

Using the frequently appearing keywords in each Swebok knowledge area and ML-related keywords, we first performed a systematic paper collection. We reviewed the collected papers and mapped the challenge topics to Swebok knowledge areas.

This paper reports the preliminary results of our work. Section 2 provides a short description of Swebok. Section 3 introduces the research method, and Section 4 reports the research results. The paper concludes by discussing the limitations of this work in Section 5.

2. The software engineering body of knowledge (Swebok)

Swebok Version 3.0 [6] is the most recently published version of the body of knowledge for the field of software engineering. Its 15 knowledge areas (KAs) summarize basic concepts and include a reference list pointing to more detailed information. The KAs are listed below:

Table 1
Search keywords extracted from Swebok3.0 KAs

Knowledge area	Keyword from KA
Software Requirements	software requirements
Software Design	software design
Software Construction	software construction
Software Testing	software testing
Software Maintenance	software maintenance
Software Configuration Management	software configuration management
Software Engineering Management	software engineering management
Software Engineering Process	software engineering process
Software Engineering Models and Methods	software engineering models and methods
Software Quality	software quality
Software Engineering Professional Practice	engineer professional
Software Engineering Economics	decision cost

Chapter 1 Software Requirements
Chapter 2 Software Design
Chapter 3 Software Construction
Chapter 4 Software Testing
Chapter 5 Software Maintenance
Chapter 6 Software Configuration Management
Chapter 7 Software Engineering Management
Chapter 8 Software Engineering Process
Chapter 9 Software Engineering Models and Methods
Chapter 10 Software Quality
Chapter 11 Software Engineering Professional Practice
Chapter 12 Software Engineering Economics
Chapter 13 Computing Foundations
Chapter 14 Mathematical Foundations
Chapter 15 Engineering Foundations

To categorize SE challenges, we consider the KAs specific to software engineering (Chapters 1–12). We do not use the KAs of Chapters 13–15, because these are also the foundational KAs for other engineering fields.

3. Research methods

3.1. Paper collection

A voluminous number of papers on machine learning and its applications have been published in many international conferences, journals and websites. This trend is continuing and may be accelerating. Researches on machine learning applications such as security, medical systems, and automated vehicles are interdisciplinary. Researches involving both ML and SE, which include a number of emerging topics, are also interdisciplinary. A literature search of specified conferences and journals on ML and SE failed to find adequate papers for our purpose; we thus designed a systematic paper collection based on the snowballing approach [7].

3.1.1. Start set

The first step generates search keywords from the frequently appearing keywords in each KA extracted by a text mining tool [8], excluding the foundational KAs (Chapters 13–15). The extracted keywords are listed in Table 1. Each keywords is the name of the corresponding KA except "Software Engineering Professional Practice" and "Software Engineering Economics". We generate search keyword pairs to use the Google search engine by combining each keyword from Swebok with the ML-related keywords "machine learning", "deep learning", and "artificial intelligence".

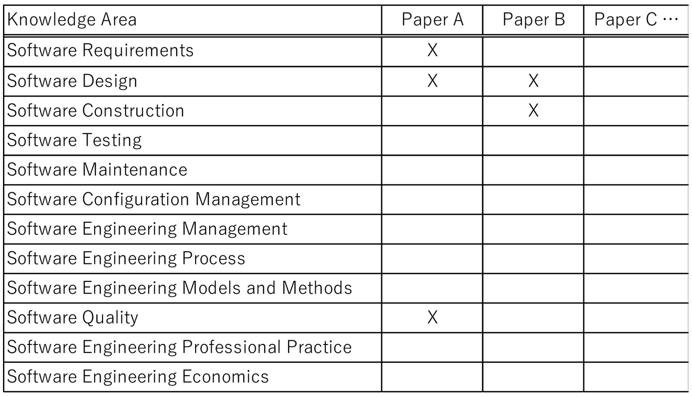
To construct the start set, we defined the following inclusion criteria for the selection of papers (websites) reported by the Google search.

- Papers that discuss or report SE challenges for ML applications, and survey software engineering techniques (e.g., software testing) for ML applications.
- Papers published in journals, proceedings of international conferences, workshops, and technical reports (including arXiv), after 2000.
- The most recent version (if multiple versions have been published).

Additional papers were collected by searching with the frequently appearing keywords in the above collected papers. These keywords were “Model engineering”, “Automated Machine Learning”, “Metamorphic testing”, and “Technical Debt”.

3.1.2. Iterations

We iteratively conducted backward and forward snowballing with the start set described in 3.1.1. By this process,



Knowledge Area	Paper A	Paper B	Paper C ···
Software Requirements	X		
Software Design	X	X	
Software Construction		X	
Software Testing			
Software Maintenance			
Software Configuration Management			
Software Engineering Management			
Software Engineering Process			
Software Engineering Models and Methods			
Software Quality	X		
Software Engineering Professional Practice			
Software Engineering Economics			

Fig. 2. The image of relation map.

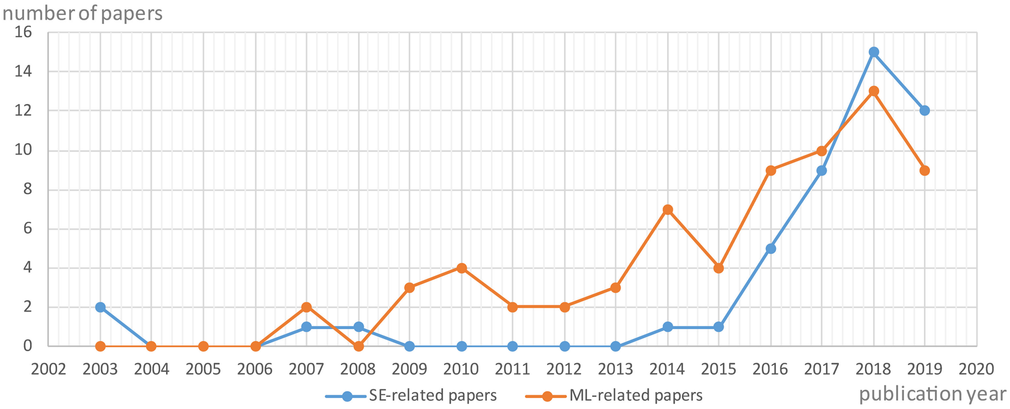


Fig. 3. The inter-annual changes of the number of selected papers within the years 2000–2019.

we additionally collected the following papers:

- Papers that overview the challenges of ML techniques.
- Survey papers on ML techniques.
- Papers that survey the challenges of ML applications.

The above inclusion criteria were necessary because we cannot discuss SE challenges without discussing the evolving ML techniques, growing application domains, and emerging ML challenges.

3.2. Challenge identification and relation mapping

After reviewing the collected papers, we identified the challenge topics from the perspectives of SE and ML, and formed a relational map between the challenges and Swebok KAs. The image of relation map is shown in Fig. 2. Paper A describes the challenges related to the software requirements, design, and quality of ML applications. Paper B reviews challenges on some kinds of learning algorithms that impact the design and construction of ML applications. These challenges are related to Software Design and Software Construction. There are papers that surveys an application domain of machine learning such as security, medical systems, and automated vehicles. However, some of survey papers describing the specific challenges of the application domain and machine learning are outside of the SE perspective. Such papers were excluded from the mapping. The mapping and challenge topics in each KA will be detailed in Section 4.

4. Research results

The literature search process in the previous section yielded 115 papers (see Table 2). SE-related papers refer to

Table 2
The numbers of selected papers

Collection phase	SE-related paper	ML-related paper
Start set	12	1
Iteration 1	23	16
Iteration 2	12	51

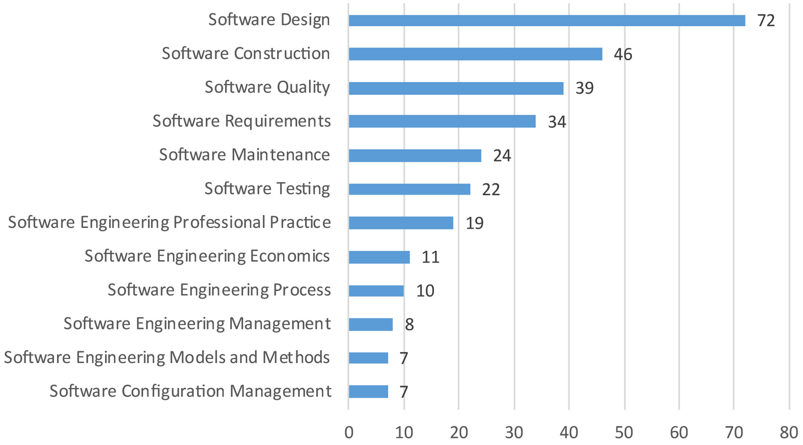


Fig. 4. The number of mapping papers to each KA.

SE challenges or survey software engineering techniques (e.g., software testing) for ML applications. ML-related papers cover challenges on ML techniques, survey papers on ML techniques, and ML applications. In the first phase of the paper collection (Start Set), we selected 13 papers: 12 SE-related papers and one ML-related paper. The ML-related paper surveyed the verification and validation of ML-base systems in the automotive industry [20]. In the following phases (Iterations 1 and 2), we selected 35 SE-related papers and 67 ML-related papers. The number of ML-related papers was significantly increased (by 51 papers) after Iteration 2, because ML techniques anchor ML applications; accordingly, SE-related papers also refer to papers on ML techniques and challenges.

Figure 3 shows the inter-annual changes in the number of selected papers in the 2000–2019 period (where the papers in 2019 were published from January to September). Note that the selected papers do not address individual techniques. The first SE-related paper after 2000 was published by Senyard et al. [13] in 2003. Few papers were published from 2004 to 2014, but significantly more were published from 2015 onward. Meanwhile, the number of ML related-papers moderately increased from 2008. This increase suggests that SE practices and their challenges for ML applications have drawn attention from research communities and practitioners since 2015. These trends are consistent with the general trends of publications on secure deep learning research [19].

Figure 4 shows the number of papers related to each KA of Swebok. Among the 115 selected papers, 108 papers were related to KAs. Software Design was related to the most number of papers in the mapping process, followed by Software Construction. Various topics were related to Software Design and Software Construction, and many survey papers addressed the ML-specific techniques and challenges of these two KAs.

In the remainder of Section 4, we will briefly describe the KAs, quoting the definitions of Swebok3.0, then overview the challenge topics in each KA.

4.1. Software requirements

4.1.1. Definition by Swebok3.0

“The Software Requirements knowledge area (KA) is concerned with the elicitation, analysis, specification, and validation of software requirements as well as the management of requirements during the whole life cycle of the software product.”

4.1.2. Challenges

Software requirements activities for ML applications involve ML-specific activities, namely, data and feasibility analysis, requirements elicitation, requirement specifications, and validation of ML-functions and performances. These activities are difficult because the requirements may change frequently in large-scale systems such as automated vehicles, they are also very complex [59,90]. Khalajzadeh et al. [25] points out “a need to better capture requirements, changes in the requirements, and adaptation of the specified process. . . . we want to better support domain expert end users in their requirements management for AI-based systems, providing approaches to capture their requirements not so much about the software solution but the domain problem, available data and business intelligence needed to solve it”. They identified a research direction in the development of tools that capture the requirements, changes in those requirements, and adaptations of specified processes.

ML techniques are widely used and are being integrated into mission critical systems; accordingly, safety, security and V&V (validation and verification) has become critical issues. Various topics on software requirements have been discussed [12,13,19,20,30,32,35,47–49,61,82,89,90,102–104,106,107,119]. Developing domain specific languages and tools for ML applications is a research direction for these challenges [12,66]. Along with safety, the interpretability of ML applications has become hotly discussed. "What is interpretability?" and "How to realize it?" are widely discussed in artificial intelligence (AI) communities (e.g., [3,51,87,104,117, 119]). Interpretability as a property of software requirements has emerged with the progress of ML techniques and applications. Fairness is another emerging property [26,118].

Requirement activities on data-oriented works have brought new challenges. Lwakatare et al. [15] and Kim et al. [40] reported the difficulty of specifying desirable datasets. Furthermore, the needs to preserve the privacy and safety of sensitive datasets and to ensure legal compliance with a new regulation such as the European General Data Protection Regulation may impact research directions in requirements engineering [18,24,60,105].

4.2. Software design

4.2.1. Definition by Swebok3.0

"A software design (the result) describes the software architecture – that is, how software is decomposed and organized into components – and the interfaces between those components. It should also describe the components at a level of detail that enables their construction."

4.2.2. Challenges

As noted in Fig. 4, many papers were related to the Software Design knowledge area. The selected papers were divided into the following categories:

- Security, Safety and V&V: Design challenges for security, safety and V&V of ML applications [12,13,19,20, 27,30,33,35,47–49,61].
- Software Structure: Challenges on the software structure of ML applications. This category includes the complex software modules of ML algorithms [14], anti-patterns in ML applications [17], and various design issues in ML models (model selection, customization and reuse) [16,34,58,59].
- Data Design: Design issues on data collection, pre-processing, cleaning, labeling and augmentation, including big data challenges [17,23,25–27,31,33,41,44,74–76,105,114–116].
- Visualization: Technical challenges on visualization techniques for the design of ML applications [25,56,63, 122,123].
- Tools: Needs of designing tools for ML applications, such as tools for non-expert ML designers, visualization tools for understanding the relationships between data and the behavior of algorithms, tools for ensuring interoperability with other tools, and domain-specific language (DSL) support [16,25,42,66].
- User Interface: Challenges on user interface design such as the interaction between users and ML applications [93–96].
- Automated ML (Auto ML): The design and construction of well-performed ML models is time consuming, requires a significant amount of resources and highly specialized experts. These demands have hindered the development of ML applications in industry. Automated machine learning (AutoML) is a new research topic that aims to resolve this problem [29,54,55,64,78,109,113].
- ML techniques (except AutoML): Brodley et al. [71] argued that application-driven research begets novel ML techniques. The contrary can also be true; that is, the challenges and solutions on ML techniques such as data processing (e.g., feature extraction) [62,70,124], ML algorithms/ models (e.g., transfer learning) [43,68,69,79,80,86,97,101,110–112,116], and specific ML functions (e.g., interpretability) [32,104,105,121] can influence the structure and implementation of ML applications.

4.3. Software construction

4.3.1. Definition by Swebok3.0

"The term software construction refers to the detailed creation of working software through a combination of coding, verification, unit testing, integration testing, and debugging. The Software Construction knowledge area (KA) is linked to all the other KAs, but it is most strongly linked to Software Design and Software Testing because the software construction process involves significant software design and testing."

4.3.2. Challenges

As mentioned above, the Software Construction KA is strongly linked to the Software Design and Software Testing KAs. When selecting papers relevant to this KA, we focused on the link between design and construction. Selected papers for this KA are also related to the Software Design KA (except Islam et al. [28], who reported the challenges facing the use of ML libraries). On the contrary, some papers related to the Software Design KA were not related to the Software Construction KA [12,13,19,20]. These papers did not discuss the challenges of constructing ML applications, but their topics were potentially closely related to construction challenges.

4.4. Software testing

4.4.1. Definition by Swebok3.0

"Software testing consists of the dynamic verification that a program provides expected behaviors on a finite set of test cases, suitably selected from the usually infinite execution domain."

4.4.2. Challenges

ML testing has drawn significant attention within the research and industrial communities because it is both important and difficult. Many researches on ML testing have been published, but many challenges remain and still emerging.

We identified ML testing challenges in the following type of papers:

- Survey papers on ML testing [52,53,57].
- Research papers on ML testing which also discuss challenges on testing [11,45,50,77].
- Survey papers on the security, safety or V&V for ML applications [12,19,30,48,107,119].
- Survey papers on the data or model management for ML applications [23,26,27].
- Papers discussing the SE challenges for ML applications [17,18,21,59,83] or the challenges in an application domain [90].

The various challenge topics on ML testing are listed below:

- Oracle Problem: How to make reliable test oracles with less human intervention for ML applications.
- Cost Reduction: Cost reduction techniques of ML testing, including the cost reduction of traditional methodologies such as search-based test-case generation, test prioritization, and test-case minimization.
- Testing ML techniques: Many of current researches focus on supervised learning. There are challenging issues on testing other ML mechanisms such as unsupervised learning, reinforce learning, transfer learning and meta-learning.
- Testing Properties: Testing of ML specific properties such as overfitting, interpretability and fairness.
- Benchmarks: The design and construction of reusable testing assets for ML applications.
- Testing Data: Testing for data validation and data cleaning. For designing secure ML applications, adversarial testing on the training data and data testing to detect privacy violations can be challenging issues.
- Mutation Testing: The design and embedding of mutants to improve simulations of real-world ML bugs.

4.5. Software maintenance

4.5.1. Definition by Swebok3.0

"In this Guide, software maintenance is defined as the totality of activities required to provide cost-effective support to software. Activities are performed during the pre-delivery stage as well as during the post-delivery stage. Pre-delivery activities include planning for post-delivery operations, maintainability, and logistics determination for transition activities. Post-delivery activities include software modification, training, and operating or interfacing to a help desk."

4.5.2. Challenges

In real-world ML applications, uncertain events might occur in the deployment phase. The environment of production ML might largely differ from the environment the ML models were trained and evaluated. In a ML applications, the ML models may be frequently retrained with concept drifts and thus change behavior autonomously in unintended ways. These situations can pose various maintenance challenges of ML applications:

- Troubleshooting: Identifying problems, diagnosing the root causes and influences of failures, and correcting faults (debugging) in the deployment phase. Automatic recovery from failures includes reconfigurations and code repairs [10,15–19,35–37,42,83].
- Runtime Monitoring: Selection of the metrics used for monitoring, live monitoring of system behavior that allows automated responses without direct human intervention, and dynamic monitoring for runtime verification and certification [17,19,24,30,47,48].
- Data Management: Tools for data dependencies, automatic data validation and cleaning during runtime, and concept drift adaptation [17,23,60,69].
- Model Management: Challenge topics on ML model management in the deployment phase, including model validation, decisions on model retraining, adversarial settings, and backwards compatibility of trained models. The governance issues in model management also fit within this category [27,34].
- Operating Environment: In ML applications, the deployment phase will most likely add new functional modules to the existing system. In the deployment and operation phases, the platform and infrastructure of the ML application might greatly differ from the training and evaluation environment of the ML model. These differences pose compatibility, portability and scalability challenges [24,59,107].

4.6. Software configuration management (SCM)

4.6.1. Definition by Swebok3.0

"Software configuration management (SCM) is a supporting-software life cycle process that benefits project management, development and maintenance activities, quality assurance activities, as well as the customers and users of the end product."

4.6.2. Challenges

To operate real-world ML applications, the complex data configuration management is indispensable as well as the software configuration management. Amershi et al. [16] points “machine learning is all about data. The amount of effort and rigor it takes to discover source, manage, and version data is inherently more complex and different than doing the same with software code.” A large-scale ML application involves a wide range of configurable objects such as the models and their options, the data and the pre- or post-processing of data [17]. As mentioned in 4.5, there are challenges on ML model management and governance [18,27,34,65,88]. Configuration management tools for ML applications should be designed in consideration of the above properties and challenges.

4.7. Software engineering management

4.7.1. Definition by Swebok3.0

"Software engineering management can be defined as the application of management activities – planning, coordinating, measuring, monitoring, controlling, and reporting – to ensure that software products and software engineering services are delivered efficiently, effectively, and to the benefit of stakeholders."

4.7.2. Challenges

This KA is concerned with topics on the software engineering project management. We selected 8 papers which include the topics and identified the following challenge issues:

- Risk Management: Risk management of the development, deployment and operation of ML applications is critical, but is rendered difficult by various uncertainties [17,24].
- Effort Estimation: Estimating the effort of an ML project is challenging because it is difficult to know to what extent the ML model will achieve its goal, and to estimate how many iterations will be needed to reach the state in which the performance gets acceptable levels [18,83].
- Corporate Compliance: In a real-world ML application project for a company, the development, deployment and operation may be severely affected by the effort of complying with the privacy policy of the organization and the legal framework. These demands impose challenges from both technical and management perspectives [16,34,60,107].

4.8. Software engineering process

4.8.1. Definition by Swebok3.0

"In this knowledge area (KA), software engineering processes are concerned with work activities accomplished by software engineers to develop, maintain, and operate software, such as requirements, design, construction, testing, configuration management, and other software engineering processes."

4.8.2. Challenges

We identified 10 papers which includes the topics on software process of ML applications. The software development lifecycle for non-ML applications is inadequate for ML applications because of the lack of consideration for data-oriented and model-oriented works including their lifecycle managements. Khomh et al. [21] posed two questions: "How should software development teams integrate the AI model lifecycle (training, testing, deploying, evolving, and so on) into their software process?" and "What new roles, artifacts, and activities come into play, and how do they tie into existing agile or DevOps processes?"

Several software processes for ML applications have been proposed [13,19,30,67,88,89]. Amershi et al. [16] discussed the process maturity model for building ML applications. Tool support for the development process can be a further challenge issue. Patel et al. [38] argues that "it is clear that non-expert tools need to support the entire exploratory and iterative process of applying statistical machine learning algorithms." Ishikawa et al. [83] reported the difficulties to make customers better understand the properties of ML applications such as imperfections. Trial-based processes can address these difficulties, but further researches are needed to build a solid foundation for the engineering disciplines.

4.9. *Software engineering models and methods*

4.9.1. *Definition by Swebok3.0*

"Software engineering models and methods impose structure on software engineering with the goal of making that activity systematic, repeatable, and ultimately more success-oriented. Using models provides an approach to problem solving, a notation, and procedures for model construction and analysis. Methods provide an approach to the systematic specification, design, construction, test, and verification of the end-item software and associated work products."

4.9.2. *Challenges*

To identify papers related to Software Engineering Models and Methods, we focused on formal methods and domain specific languages. Hains et al. [12] proposed research directions: Domain-specific languages (DSL) and tools for a formal specification, UML class diagrams for representing datasets, model-based testing tools and theorem-proving techniques. Portugal et al. [66] briefly surveyed DSL for machine learning in Big Data. They reported "no DSL was found that targeted the expression of systems requirements". The remaining papers related to this KA also discussed the challenges on the formal approach of ML techniques [13,47–49,102].

4.10. *Software quality*

4.10.1. *Definition by Swebok3.0*

The Swebok Guide asks "What is software quality, and why is it so important that it is included in many knowledge areas (KAs) of the SWEBOK Guide?" Actually, software quality is an umbrella term for multiple facets. It refers to whether the software products possess the desired characteristics, the extent to which a software product possesses those characteristics, and the processes, tools, and techniques by which the developer achieves those characteristics. Throughout its history, the term software quality has been differently defined by researchers and organizations. The Software Quality KA provides definitions and "the practices, tools, and techniques for defining software quality and for appraising the state of software quality during development, maintenance, and deployment."

4.10.2. *Challenges*

The Software Quality KA broadly covers topics on software quality. The software quality challenges in ML applications also embrace various topics. The representative challenge is software testing, which is excluded here because it was discussed in Section 4.4. The other challenge topics in software quality are listed below:

- Quality Assurance: Software quality assurance is "a set of activities that define and assess the adequacy of software processes." It confirms that the software processes can complete the target task and that the software products fulfil their intended purposes [6]. Some papers have discussed the question "What is the adequate quality assurance for ML applications? and how to perform it?" [13,19,21,30,61,81,82,85,89,90,102].
- Validation & Verification: V&V is the integral part of software quality assurance for ML applications. Many papers addressed technical challenges of V&V for ML applications [13,30,47–49,90,102,119].

- Fault Analysis: Trouble shooting issues of ML applications such as fault characterization, detection and elimination [10,22,36,37,122] (see also 4.5).
- Component Quality: Training data, ML models and ML platforms (e.g. scikit-learn [125], Tensor flow [126], Weka [127]) is the key components of ML applications. The challenges on the quality of each components have been discussed. The challenge issues on data quality for ML applications includes data anomaly [23], imbalanced or biased data, encrypted data [60], data evaluation/cleaning [39–41,74,114,115], novelty detection [31]. The quality of ML model and ML platform are mainly discussed in the context of testing (see Section 4.5).
- Quality Measurement: Breck et al. [45] suggested a quality measure of ML applications. They proposed a test scoring method to measure the production readiness of a given ML application. Quality measurement for ML applications is closely related to system safety. Varshney et al. [103] discussed the definition of safety from the view of reduction or minimization of risk and epistemic uncertainty associated with unwanted outcomes that are severe enough to be seen as harmful. The measures for risk and uncertainty of ML applications can also become safety measures. Corbett-Davies et al. [118] proposed a measure for evaluating fairness of ML applications.
- Safety and Security: Mission critical systems in some domains are strongly required safety and security. There are industry standards which address safety or security such as DO-178 [128], ISO26262 and ASIL [129] and Common Criteria [130]. How to conform ML applications to these standards is difficult but important challenges to realize mission critical ML applications in industry [20,30,35,61,82,90,119].
- Ethics and Regulations: The ethics of ML applications relate to issues on safety, privacy and discrimination [20, 24,49,59,103,106]. Ethical topics should be included in quality evaluations of ML applications. Some papers have discussed the impact of regulations on the development/deployment of ML applications [20,24,49,87,103,105]. The imposed regulations will also affect the quality of ML applications.

4.11. Software engineering professional practice

4.11.1. Definition by Swebok3.0

"The Software Engineering Professional Practice knowledge area (KA) is concerned with the knowledge, skills, and attitudes that software engineers must possess to practice software engineering in a professional, responsible, and ethical manner."

4.11.2. Challenges

The lifecycle process of ML applications includes wide range of works such as data analysis, data pre-processing, data cleaning, ML model design and construction, system deployment and operations (e.g. monitoring, debugging and retraining). The skills needed for these works may go far beyond the scope of traditional software engineering. Developing the skill set for ML applications is the major challenge related to this KA [15,16,27,32,39,40,59]. The other topics related to this KA were identified as follows:

- Group Dynamics and Psychology: To construct and efficiently deploy a high-performance ML application, various stakeholders with different knowledge sets, skills and cultures should participate in the project. The main challenges in this category are collaboration to ensure a successful project and adequate communication with customers [17,18,21,27,40,59,83].
- Economic Impacts: The business impact of real-world ML applications is a crucial factor. ML applications engineers should possess the techniques and skills to analyze the business impacts (see also Section 4.12).
- Ethics and Regulations: The ethics and regulation of ML applications also relate to the professionalism of ML applications engineers (see Section 4.10).

4.12. Software engineering economics

4.12.1. Definition by Swebok3.0

"This knowledge area (KA) provides an overview on software engineering economics. Economics is the study of value, costs, resources, and their relationship in a given context or situation. In the discipline of software engineering, activities have costs, but the resulting software itself has economic attributes as well. Software engineering economics provides a way to study the attributes of software and software processes in a systematic way that relates them to economic measures."

4.12.2. Challenges

Software engineering economics is crucial for read-world ML applications in industry. Besides economic topics, this KA covers risk and uncertainty management. However, very few of our collected papers discussed the challenges related to this KA. These challenges can be divided into two categories:

- Risk and Uncertainty management: Technical debts which may result in maintenance cost escalation [17]. Other challenges in this category are project risk estimation [24,107], and difficulties in estimating the effort arising from uncertainties in ML applications [18,83,92].
- Economic Impact: Lucas et al. [60] reported the difficulties of translating ML results into real business impacts.

Most of the performance metrics on ML techniques are not easily understood by customers, who are expected to be unfamiliar with ML techniques. Therefore, customers cannot easily translate metrics such as accuracy into relevant key performance indicators such as revenue. Dahlmeier [84] highlighted challenges that make it difficult to translate the results into impactful innovation in natural language processing (NLP) research. They pointed out "lack of value focus" in the current NLP researches. The same problem may exist in ML researches. As another research priorities, Russell et al. [49] discussed optimizing AI's economic impact which includes labor market forecasting, market disruptions through the use of AI techniques and policy for managing adverse effects.

5. Conclusion

In this review, we attempted to broadly outline the SE challenges for ML applications by a systematic review and mapping them to knowledge areas (KAs) in Swebok3.0. As the result, 115 papers were selected by our systematic collection. Among them, 108 papers were mapped to KAs of Swebok. The remaining seven papers surveyed ML applications of some specific domain such as robotics [91,98–100], medical systems [46], networking [72] and machine translation [120]. They mainly discussed on domain specific challenges; their challenges and solutions may be potentially related to software engineering activities for such domain applications.

The broad range of challenge topics were extracted through the mapping. They were related to several KAs. In particular, safety, security and V&V for ML applications are major challenge topics over several KAs. We also identified challenge topics for engineering practice such as ethic and regulations, economic impacts and risk managements.

The research method designed for our purpose is based on the existing systematic methods [7,9]. This paper reports the results of two iterations of the snowballing approach for paper collection. We believe that the collection result is comprehensive to some extent, but that more iterations would provide more comprehensive results. Note that even the most comprehensive collection would provide only a snapshot because related papers are published daily. In our backward and forward snowballings, the additional papers to include were selected by one researcher. The relation mapping was also conducted by the same researcher. To achieve more objective and persuasive conclusions, multiple persons must review, select and map the included papers. Threats to the validity of this method must also be carefully discussed.

Although the current results are preliminary and subjected to the above limitations, we expect that they will help to elucidate the whole aspect of SE challenges for ML applications.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP19K03011.

References

[1] The Software Engineering for Machine Learning Applications (SEMLA) international symposium, [homepage on the Internet]. 2019 [cited September 2019]. Available from: https://semla.polymtl.ca/.

[2] The Conference on Systems and Machine Learning (SysML), [homepage on the Internet]. 2019 [cited September 2019]. Available from: http://www.sysml.cc/.

[3] International Workshop on Data Management for End-to-End Machine Learning (DEEM), [homepage on the Internet]. 2019 [cited September 2019]. Available from: http://
deem-workshop.org/index.html.

[4] Kläs M, Vollmer AM. Uncertainty in Machine Learning Applications: A Practice-Driven Classification of Uncertainty. In: Gallina B, Skavhaug A, Schoitsch E, Bitsch F, eds. Computer Safety, Reliability, and Security. SAFECOMP 2018. Lecture Notes in Computer Science, vol 11094. Springer, Cham, 2018.

[5] Faria JM. Non-determinism and Failure Modes in Machine Learning, 2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Toulouse, 2017, pp. 310-316.

[6] Bourque P, Fairley RE, eds. Guide to the Software Engineering Body of Knowledge, Version 3.0, IEEE Computer Society, [homepage on the Internet]. 2014 [cited September 2019]. Available from: www.swebok.org.

[7] Wohlin C. Guidelines for snowballing in systematic literature studies and a replication in software engineering, Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, ACM, 2014.

[8] KH Conder. [homepage on the Internet]. [cited September 2019]. Available from: https://khcoder.net/en/.

[9] Petersen K, Vakkalanka S, Kuzniarz L. Guidelines for conducting systematic mapping studies in software engineering: An update, Information and Software Technology, 64, 2015, 1-18.

[10] Sun X, Zhou T, Li G, Hu J, Yang H, Li B. An Empirical Study on Real Bugs for Machine Learning Programs, 2017 24th Asia-Pacific Software Engineering Conference (APSEC), Nanjing, 2017, pp. 348-357.

[11] Sekhon J, Fleming C. Towards improved testing for deep learning. In Proceedings of the 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER '19), 2019.

[12] Hains G, Jakobsson A, Khmelevsky Y. Towards formal methods and software engineering for deep learning: Security, safety and productivity for dl systems development, 2018 Annual IEEE International Systems Conference (SysCon), Vancouver, BC, 2018, pp. 1-5.

[13] Senyard A, Kazmierczak E, Sterling L. Software engineering methods for neural networks, Tenth Asia-Pacific Software Engineering Conference, 2003, Chiang Mai, Thailand, 2003, pp. 468-477.

[14] Kriens P, Verbelen T. Software Engineering Practices for Machine Learning. arXiv preprint arXiv:1906.10366, 2019.

[15] Lwakatare LE, Raj A, Bosch J, Olsson HH, Crnkovic I. A Taxonomy of Software Engineering Challenges for Machine Learning Systems: An Empirical Investigation. In: Kruchten P, Fraser S, Coallier F, eds. Agile Processes in Software Engineering and Extreme Programming. XP 2019. Lecture Notes in Business Information Processing, vol 355. Springer, Cham, 2019.

[16] Amershi S, Bird C, DeLine R, Gall H, Kamar E, Nagappan N, Nushi B, Zimmermann T. Software engineering for machine learning: a case study. In Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP '10).

[17] Sculley D, Holt G, Golovin D, Davydov E, Phillips T, Ebner D, Chaudhary V, Young M, Crespo JF, Dennison D. Hidden technical debt in Machine learning systems. In Proceedings of the 28th International Conference on Neural Information Processing Systems – Volume 2 (NIPS'15), Vol. 2. MIT Press.

[18] Anders A, et al. Software engineering challenges of deep learning. 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), IEEE, 2018.

[19] Lei M, et al. Secure Deep Learning Engineering: A Software Quality Assurance Perspective. arXiv preprint arXiv:1810. 04538, 2018.

[20] Markus B, et al. Safely entering the deep: A review of verification and validation for machine learning and a challenge elicitation in the automotive industry. arXiv preprint arXiv:1812.05389, 2018.

[21] Khomh F, Adams B, Cheng J, Fokaefs M, Antoniol G. Software Engineering for Machine-Learning Applications: The Road Ahead, in IEEE Software, 35(5), September/October 2018, 81-84.

[22] Masuda S, Ono K, Yasue T, Hosokawa N. A Survey of Software Quality for Machine Learning Applications, 2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Vasteras, 2018, pp. 279-284.

[23] Foorthuis R. A Typology of Data Anomalies. In: Medina J, et al. eds. Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations. IPMU 2018. Communications in Computer and Information Science, vol 854. Springer, Cham, 2018.

[24] Flaounas I. Beyond the technical challenges for deploying Machine Learning solutions in a software company. arXiv preprint arXiv:1708.02363, 2017.

[25] Khalajzadeh H, Abdelrazek M, Grundy J, Hosking J, He Q. A Survey of Current End-User Data Analytics Tool Support, 2018 IEEE International Congress on Big Data (BigData Congress), San Francisco, CA, 2018, pp. 41-48.

[26] Polyzotis N, Roy S, Whang SE, Zinkevich M. Data lifecycle challenges in production machine learning: A survey, SIGMOD Rec, 47(2), December 2018, 17-28.

[27] Sebastian S, et al. On challenges in machine learning model management, IEEE Data Eng. Bull, 41(4), 2018, 5-15.

[28] Islam MJ, et al. What Do Developers Ask About ML Libraries? A Large-scale Study Using Stack Overflow. arXiv preprint arXiv:1906.11940, 2019.

[29] Lee DJL, et al. A Human-in-the-loop Perspective on AutoML: Milestones and the Road Ahead, Data Engineering, 2019, 58.

[30] Schumann J, Gupta P, Liu Y. Application of Neural Networks in High Assurance Systems: A Survey. In: Schumann J, Liu Y, eds. Applications of Neural Networks in High Assurance Systems. Studies in Computational Intelligence, vol 268. Springer, Berlin, Heidelberg, 2010.

[31] Pimentel MAF, et al. Review: A review of novelty detection, Signal Process, 99, June 2014, 215-249.

[32] Weld DS, Bansal G. The challenge of crafting intelligible intelligence, Commun. ACM, 62(6), May 2019, 70-79. doi: 10.1145/3282486.

[33] Wuest T, Weimer D, Irgens C, Thoben KD. Machine learning in manufacturing: Advantages, challenges, and applications, Production & Manufacturing Research, 4(1), 2016, 23-45.

[34] Sridhar V, et al. Model governance: Reducing the anarchy of production ML. 2018 USENIX Annual Technical Conference (USENIX ATC '18), 2018.

[35] Salay R, Queiroz R, Czarnecki K. An analysis of ISO 26262: Using machine learning safely in automotive software. arXiv preprint arXiv:1709.02435, 2017.

[36] Andrist S, Bohus D, Kamar E, Horvitz E. What Went Wrong and Why? Diagnosing Situated Interaction Failures in the Wild. In: Kheddar A. et al. eds. Social Robotics. ICSR 2017. Lecture Notes in Computer Science, vol 10652. Springer, Cham, 2017.

[37] Nushi B, et al. On human intellect and machine failures: Troubleshooting integrative machine learning systems. Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[38] Patel K, Fogarty J, Landay JA, Harrison B. Investigating statistical machine learning as a tool for software development. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08), 2008.

[39] Kim M, et al. Data scientists in software teams: State of the art and challenges, IEEE Transactions on Software Engineering, 44(11), 2017, 1024-1038.

[40] Kim M, et al. The emerging role of data scientists on software development teams. Proceedings of the 38th International Conference on Software Engineering. ACM, 2016.

[41] Polyzotis N, Roy S, Whang SE, Zinkevich M. Data Management Challenges in Production Machine Learning. In Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD '17), 2017.

[42] Hill C, Bellamy R, Erickson T, Burnett M. Trials and tribulations of developers of intelligent systems: A field study, 2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Cambridge, 2016, pp. 162-170.

[43] Attenberg J, Provost F. Inactive learning?: Difficulties employing active learning in practice, SIGKDD Explor. Newsl, 12(2), March 2011, 36-41.

[44] Chen XW, Lin X. Big data deep learning: Challenges and perspectives, in IEEE Access, 2, 2014, 514-525.

[45] Breck E, et al. What's your ML Test Score? A rubric for ML production systems. [journal on the Internet] 2016 Dec [cited September 2019]. Available from: https://ai.google/research/pubs/pub45742.

[46] Menasalvas E, Gonzalo-Martin C. Challenges of Medical Text and Image Processing: Machine Learning Approaches. In: Holzinger A, ed. Machine Learning for Health Informatics. Lecture Notes in Computer Science, vol 9605. Springer, Cham, 2016.

[47] Wesel P. Challenges in the Verification of Reinforcement Learning Algorithms, NASA/TM-2017-219628, [journal on the Internet] 2017 Jun [cited September 2019]. Available from: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170007190.pdf.

[48] Taylor B, Darrah M, Moats C. Verification and validation of neural networks: a sampling of research in progress, Proc. SPIE 5103, Intelligent Computing: Theory and Applications, 2003.

[49] Russell S, Dewey D, Tegmark M. Research priorities for robust and beneficial artificial intelligence, Ai Magazine, 36(4), 2015, 105-114.

[50] Leofante F, Pulina L, Tacchella A. Learning with Safety Requirements: State of the Art and Open Questions. RCRA@ AI* IA. 2016.

[51] Doshi-Velez F, Kim B. Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608, 2017.

[52] Zhang JM, et al. Machine Learning Testing: Survey, Landscapes and Horizons. arXiv preprint arXiv:1906.10742, 2019.

[53] Braiek HB, Khomh F. On testing machine learning programs. arXiv preprint arXiv:1812.02257, 2018.

[54] Yao Q, et al. Taking human out of learning applications: A survey on automated machine learning. arXiv preprint arXiv: 1810.13306, 2018.

[55] Zöller MA, Huber MF. Survey on Automated Machine Learning. arXiv preprint arXiv:1904.12054, 2019.

[56] Garcia R, et al. A task-and-technique centered survey on visual analytics for deep learning model engineering, Computers & Graphics, 77, 2018, 30-49,

[57] Salman S, et al. A Systematic Mapping Study on Testing of Machine Learning Programs. arXiv preprint arXiv:1907.09427, 2019.

[58] Ghofrani J, et al. Reusability in Artificial Neural Networks: An Empirical Study. In Proceedings of the 23rd International Systems and Software Product Line Conference – Volume B (SPLC '19).

[59] Rahman MS, et al. Machine Learning Software Engineering in Practice: An Industrial Case Study. arXiv preprint arXiv: 1906.07154, 2019.

[60] Lucas B, Fabian J, Stefan S. Challenges in the deployment and operation of machine learning in practice. In Proceedings of the 27th European Conference on Information Systems (ECIS), Stockholm & Uppsala, Sweden, June 8–14, 2019.

[61] Kuwajima H, Yasuoka H, Nakae T. Open Problems in Engineering Machine Learning Systems and the Quality Model. arXiv preprint arXiv:1904.00001, 2019.

[62] Cunningham JP, Ghahramani Z. Linear dimensionality reduction: Survey, insights, and generalizations, The Journal of Machine Learning Research, 16(1), 2015, 2859-2900.

[63] Liu S, et al. Visualizing high-dimensional data: advances in the past decade, in IEEE Transactions on Visualization and Computer Graphics, 23(3), 2017, 1249-1268.

[64] Gil Y, et al. Towards human-guided machine learning. In Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI '19), 2019.

[65] Garcia R, et al. Context: The missing piece in the machine learning lifecycle, KDD CMI Workshop, 114, 2018.

[66] Portugal I, Alencar P, Cowan D. A Preliminary Survey on Domain-Specific Languages for Machine Learning in Big Data, 2016 IEEE International Conference on Software Science, Technology and Engineering (SWSTE), Beer-Sheva, 2016, pp. 108-110.

[67] Falcini F, Lami G. Deep Learning in Automotive: Challenges and Opportunities. In: Mas A, ed. Software Process Improvement and Capability Determination. SPICE 2017. Communications in Computer and Information Science, vol 770. Springer, Cham, 2017.

[68] Arlot S, Celisse A. A survey of cross-validation procedures for model selection, Statistics Surveys, 4, 2010, 40-79. doi: 10.1214/09-SS054.

[69] Gama J, et al. A survey on concept drift adaptation, ACM Computing Surveys, 46(4), March 2014. doi: 10.1145/2523813.

[70] Li J, et al. Feature selection: A data perspective, ACM Computing Surveys (CSUR), 50(6), 2018, 94. doi: 10.1145/3136625.

[71] Brodley CE, Rebbapragada U, Small K, Wallace BC. Challenges and opportunities in applied machine learning, Ai Magazine, 33(1), 2012, 11-24.

[72] Boutaba R, et al. A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities, Journal of Internet Services and Applications, 9, 2018, 1-99.

[73] Bibal A, Frénay B. Interpretability of machine learning models and representations: an introduction. 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, 2016, pp. 77-82.

[74] Roh Y, et al. A Survey on Data Collection for Machine Learning: a Big Data – AI Integration Perspective. ArXiv abs/1811.03402, 2018.

[75] Ramrez-Gallego S, et al. A survey on data preprocessing for data stream mining, Neurocomput, 239(C), May 2017, 39-57. doi: 10.1016/j.neucom.2017.01.078.

[76] Storcheus D, Rostamizadeh A, Kumar S. A survey of modern questions and challenges in feature extraction. The 1st International Workshop "Feature Extraction: Modern Questions and Challenges". 2015.

[77] Bunel R, et al. Piecewise Linear Neural Network verification: A comparative study. ArXiv abs/1711.00455, 2018.

[78] He X, Zhao K, Chu X. AutoML: A Survey of the State-of-the-Art. arXiv preprint arXiv:1908.00709, 2019.

[79] Heda S, et al. A review on the self and dual interactions between machine learning and optimization, Progress in Artificial Intelligence, 8, 2019, 143-165.

[80] Wang Y, et al. Generalizing from a Few Examples: A Survey on Few-Shot Learning. arXiv preprint arXiv:1904.05046, 2019.

[81] Ashmore R, Calinescu R, Paterson C. Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges. arXiv preprint arXiv:1905.04223, 2019.

[82] Faria JM. Machine learning safety: An overview. Proceedings of the 26th Safety-Critical Systems Symposium, York, UK, 2018.

[83] Ishikawa F, Yoshioka N. How do engineers perceive difficulties in engineering of machine-learning systems?: questionnaire survey. In Proceedings of the Joint 7th International Workshop on Conducting Empirical Studies in Industry and 6th International Workshop on Software Engineering Research and Industrial Practice (CESSER-IP '19), 2019.

[84] Dahlmeier D. On the Challenges of Translating NLP Research into Commercial Products. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2017.

[85] Ishikawa F, Matsuno Y. Continuous Argument Engineering: Tackling Uncertainty in Machine Learning Based Systems. In: Gallina B, Skavhaug A, Schoitsch E, Bitsch F, eds. Computer Safety, Reliability, and Security. SAFECOMP 2018. Lecture Notes in Computer Science, vol 11094. Springer, Cham, 2018.

[86] Liu W, et al. A survey of deep neural network architectures and their applications, Neurocomputing, 234, 2017, 11-26.

[87] Lipton ZC. The mythos of model interpretability. arXiv preprint arXiv:1606.03490, 2016.

[88] Miao H, Li A, Davis LS, Deshpande A. Towards Unified Data and Lifecycle Management for Deep Learning, 2017 IEEE 33rd International Conference on Data Engineering (ICDE), San Diego, CA, 2017, pp. 571-582.

[89] Kurd Z, Kelly T, Austin J. Developing artificial neural networks for safety critical systems, Neural Computing & Applications, 16(1), 2007 Jan, 11-19.

[90] Koopman P, Wagner M. Challenges in autonomous vehicle testing and validation, SAE Int. J. Trans. Safety, 4(1), 2016, 15-24,

[91] Skantze G, et al. Furhat at Robotville?: A Robot Head Harvesting the Thoughts of the Public through Multi-party Dialogue. International Conference on Intelligent Virtual Agents; 2012.

[92] Begel A, Zimmermann T. Analyze this! 145 questions for data scientists in software engineering. In Proceedings of the 36th International Conference on Software Engineering (ICSE 2014), 2014.

[93] Tullio J, Dey AK, Chalecki J, Fogarty J. How it works: a field study of non-technical users interacting with an intelligent system. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07), 2007.

[94] Stumpf S, Rajaram V, Li L, Wong W-K, Burnett M, Dietterich T, Sullivan E, Herlocker J. Interacting meaningfully with machine learning systems: Three experiments, Int. J. Hum.-Comput. Stud, 67(8), August 2009, 639-662.

[95] Stumpf S, et al. Toward harnessing user feedback for machine learning. In Proceedings of the 12th International Conference on Intelligent User Interfaces (IUI '07), 2007.

[96] Amershi S, Cakmak M, Knox WB, Kulesza T. Power to the people: The role of humans in interactive machine learning, AI Magazine, 35(4), 2014, 105-120.

[97] Pan SJ, Yang Q. A survey on transfer learning, in IEEE Transactions on Knowledge and Data Engineering, 22(10), Oct. 2010, 1345-1359.

[98] Kober J, et al. Reinforcement Learning in Robotics: A Survey. In: Learning Motor Skills. Springer Tracts in Advanced Robotics, vol 97. Springer, Cham, 2014.

[99] Nguyen-Tuong D, Peters J. Model learning for robot control: A survey, Cognitive Processing, 12(4), 2011, 319-340.

[100] Argall BD, Chernova S, Veloso M, Browning B. A survey of robot learning from demonstration, Robotics and Autonomous Systems, 57(5), 2009, 469-483.

[101] Settles B. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin-Madison. [journal on the Internet] 2009 Jan [cited September 2019]. Available from: https://research.cs.wisc.edu/techrepo rts/2009/TR1648.pdf.

[102] Pathak S, et al. How to Abstract Intelligence? (If Verification Is in Order). AAAI Fall Symposia, 2013.

[103] Varshney KR, Alemzadeh H. On the safety of machine learning: Cyber-physical systems, decision sciences, and data products, Big Data, 5(3), 2017, 246-255.

[104] Otte C. Safe and Interpretable Machine Learning: A Methodological Review. In: Moewes C, Nürnberger A, eds. Computational Intelligence in Intelligent Data Analysis. Studies in Computational Intelligence, vol 445. Springer, Berlin, Heidelberg, 2013.

[105] Goodman B, Flaxman S. European Union regulations on algorithmic decision-making and a right to explanation, Ai Magazine, 38(3), 2017, 50-57.

[106] Bostrom N, Yudkowsky E. The ethics of artificial intelligence, The Cambridge Handbook of Artificial Intelligence, 316, 2014, 334.

[107] Amodei D, et al. Concrete problems in AI safety. arXiv preprint arXiv:1606.06565, 2016.

[108] Elsken T, Metzen JH, Hutter F. Neural architecture search: A survey. arXiv preprint arXiv:1808.05377, 2018.

[109] Feurer M, Hutter F. Hyperparameter Optimization. In: Hutter F, Kotthoff L, Vanschoren J, eds. Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Springer, Cham, 2019.

[110] Pan SJ, Yang Q. A survey on transfer learning, in IEEE Transactions on Knowledge and Data Engineering, 22(10), Oct. 2010, 1345-1359.

[111] Lemke C, et al. Metalearning: A survey of trends and technologies, Artificial Intelligence Review, 44, 2015, 117-130. doi: 10.1007/s10462-013-9406-y.

[112] Vanschoren J. Meta-learning: A survey. arXiv preprint arXiv:1810.03548, 2018.

[113] Luo G. A review of automatic selection methods for machine learning algorithms and hyper-parameter values, Network Modeling Analysis in Health Informatics and Bioinformatics, 5, 2016, 1-16.

[114] Chu X, et al. Data Cleaning: Overview and Emerging Challenges. SIGMOD Conference, 2016.

[115] Qi Z, et al. Impacts of dirty data: and experimental evaluation. arXiv preprint arXiv:1803.06071, 2018.

[116] Weiss K, Khoshgoftaar TM, Wang D. A survey of transfer learning, J Big Data, 3(9), 2016. doi: 10.1186/s40537-016-0043-6.

[117] Biran O, Cotton C. Explanation and justification in machine learning: A survey. IJCAI-17 Workshop on Explainable AI (XAI). Vol. 8. 2017.

[118] Corbett-Davies S, Goel S. The measure and mismeasure of fairness: A critical review of fair machine learning. arXiv preprint arXiv:1808.00023, 2018.

[119] Huang X, et al. Safety and Trustworthiness of Deep Neural Networks: A Survey. arXiv preprint arXiv:1812.08342, 2018.

[120] Zhang J, Zong C. Deep neural networks in machine translation: An overview, in IEEE Intelligent Systems, 30(5), Sept. -Oct. 2015, 16-25.

[121] Zhang L, Wang S, Liu B. Deep learning for sentiment analysis: A survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), 2018, e1253.

[122] Hohman F, et al. Visual analytics in deep learning: An interrogative survey for the next frontiers. IEEE transactions on visualization and computer graphics, 2018.

[123] Seifert C, et al. Visualizations of Deep Neural Networks in Computer Vision: A Survey. In: Cerquitelli T, Quercia D, Pasquale F, eds. Transparent Data Mining for Big and Small Data. Studies in Big Data, vol 32. Springer, Cham, 2017.

[124] Sorzano COS, Vargas J, Pascual-Montano A. A survey of dimensionality reduction techniques. arXiv preprint arXiv: 1403.2877, 2014.

[125] scikit-learn, [homepage on the Internet]. [cited September 2019]. Available from: https://scikit-learn.org/stable/.

[126] TensorFlow, [homepage on the Internet]. [cited September 2019]. Available from: https://www.tensorflow.org/.

[127] Weka, [homepage on the Internet]. [cited September 2019]. Available from: https://www.cs.waikato.ac.nz/ml/weka/.

[128] DO-178C, Software Considerations in Airborne Systems and Equipment Certification [homepage on the Internet]. 2012 [cited September 2019]. Available from: http://www.
rtca.org.

[129] ISO/TC 22/SC 32, ISO26262 Road vehicles – Functional safety, [homepage on the Internet]. 2018 [cited September 2019]. Available from: https://www.iso.org/ics/43.040.10/x/.

[130] Common Criteria Recognition Arrangement, Common Criteria for Information Technology Security Evaluation, [homepage on the Internet]. 2017 [cited September 2019]. Available from: https://www.commoncriteriaportal.org/cc/.