Guidelines for Quality Assurance of Machine Learning-based Artificial Intelligence

Koichi Hamada
DeNA Co., Ltd
Tokyo, Japan
koichi.hamada@dena.com

Fuyuki Ishikawa
National Institute of Informatics
Tokyo, Japan
f-ishikawa@nii.ac.jp

Satoshi Masuda
IBM Research
Tokyo, Japan
smasuda@jp.ibm.com

Mineo Matsuya
LIFULL Co., Ltd.
Tokyo, Japan
matsuyamineo@lifull.com

Tomoyuki Myojin*
Japan Aerospace Exploration Agency
Tsukuba, Japan
myojin.tomoyuki@jaxa.jp

Yasuharu Nishi
University of Electro-Communications
Tokyo, Japan
Yasuharu.Nishi@uec.ac.jp

Hideto Ogawa
Hitachi, Ltd.
Yokohama, Japan
hideto.ogawa.cp@hitachi.com

Takahiro Toku
OMRON Corporation
Kyoto, Japan
takahiro.toku@omron.com

Susumu Tokumoto
FUJITSU LABORATORIES LTD.
Kawasaki, Japan
tokumoto.susumu@fujitsu.com

Kazunori Tsuchiya
FUJITSU LTD.
Kawasaki, Japan
ktsuchiya@fujitsu.com

Yasuhiro Ujita
OMRON Corporation
Kyoto, Japan
yasuhiro.ujita@omron.com

***Abstract*—Great efforts are currently underway to develop industrial applications for artificial intelligence (AI), especially those using machine learning (ML) techniques. Despite the intensive support for building ML applications, there are still challenges when it comes to evaluating, assuring, and improving the quality or dependability. The difficulty stems from the unique nature of ML: namely, that the system behavior is derived from training data, not from logical design by human engineers. This leads to black-box and intrinsically imperfect implementations that invalidate many of the existing principles and techniques in traditional software engineering. In light of this situation, the Japanese industry has jointly worked on a set of guidelines for the quality assurance of AI systems (in the QA4AI consortium) from the viewpoint of traditional quality-assurance engineers and test engineers. We report the initial version of these guidelines, which cover a list of the quality evaluation aspects, a catalogue of current state-of-the-art techniques, and domain-specific discussions in four representative domains. The guidelines provide significant insights for engineers in terms of methodologies and designs for tests driven by application-specific requirements.**

***Index Terms*—software quality, testing, artificial intelligence, machine learning, guidelines**

I. Introduction

Machine learning (ML) is a key driving force for industrial innovation in the form of artificial intelligence (AI) systems. ML-based AI systems consistently display unique characteristics in engineering because components (models) are constructed by training with data in an inductive manner. The obtained components are intrinsically imperfect, i.e., they tend to have limited accuracy, and they are black-box in the sense that the learned behavior is too complex to understand or reason about, especially in the case of deep learning. Further difficulties emerge as such AI systems work with fuzzy requirements regarding human perception or the open real world. One survey showed that more than 40% engineers feel the difficulty of quality assurance for AI systems is at the highest level in the sense that existing approaches are no longer working [1].

At the same time, there is an increasing demand for high-quality and dependable AI systems because they work closely with humans. It is therefore crucial to provide clear guidance for understanding and tackling the difficulties inherent in high-quality AI systems. In response to such industry demands, we established the Consortium of Quality Assurance for Artificial Intelligence-based products and services (QA4AI Consortium), made up of experts from both industry and academia. The objectives of the consortium are to form a societal consensus on quality of AI systems by researching issues and solutions relating to them, and to contribute to the diffusion of ML developments into a safe and secure society.

In this paper, we report the first version of the guidelines for the quality assurance of ML-based AI systems [2]. These guidelines define the general concept and technologies for the quality assurance of AI systems including concrete guidelines relating to the quality characteristics, test architecture, and test viewpoints in each typical domain.

The remainder of this paper is organized as follows: In Section II, we first describe the consortium and the methodology to work on the guidelines. In Sections III and IV, we describe the guidelines in terms of the common core part and domain-specific parts, respectively. We evaluate the

*Presently, the author is with Hitachi, Ltd.
E-mail: tomoyuki.myojin.fs@hitachi.com

DOI reference number:10.18293/SEKE2020-094

guidelines in section V, and discuss the threats to validity of the evaluation in section VI. Section VII introduces the related work of this paper. We conclude the paper with future perspective in Section VIII.

II. Methodology

A. The QA4AI Consortium

The QA4AI Consortium is a voluntary group to discuss the quality assurance of ML-based AI systems in Japan. Its objectives are to promote the application of ML-based AI systems by reducing the risks associated with AI/ML and to foster common social understanding of their quality, including limitations.

When the first version of the guidelines was released, the consortium consisted of 39 experts and three organizations from both academia and industry. Members include researchers and practitioners in various technical fields including software engineering, system safety, machine learning, and quality assurance. The application domains of the participants are also diverse, covering the entertainment, automotive, factory automation, electrics and electronics, communications, software, IT solutions, consumer devices, web systems, aerospace and more.

B. Structure of Guidelines

The consortium facilitated two types of discussion to formulate the guidelines. In the first, quality assurance-related issues in specific application domains were discussed. The purpose was to derive concrete insights, as general insights might be too abstract for the various domains with different demands. For the first version of the guidelines, there were four working groups: one each for generative systems, operational data in process systems, voice user interface, and autonomous driving.

The second type of discussion was for organizing and summarizing the common core concepts of the quality assurance of ML-based AI systems. These discussions were facilitated by expert members and their output was reviewed by the entire consortium. The common concepts consist of two parts: axes of quality evaluation and a technical catalogue.

The first version of the guidelines was published on the QA4AI Consortium's web site[1] in May 2019. It has the structure below corresponding to the two discussion types:

- Core parts of guidelines, including (1) Axes of Quality Evaluation and (2) Technical Catalogue
- Guidelines for specific domains for (1) Generative Systems, (2) Operational Data in Process Systems, (3) Voice User Interface, and (4) Autonomous Driving

III. Core Parts of Guidelines

A. Axes of Quality Evaluation

The quality assurance of ML-based systems has unique aspects in contrast to the quality assurance of traditional, non ML-based systems. Specifically, ML-based systems usually include a complex, nonlinear model constructed in the inductive development style for stakeholders, who may be unfamiliar with ML-based system development.

Software development can be divided into the deductive style and the inductive style. The former is that, for traditional software, engineers have rich knowledge on development from their experiences. Quality assurance applies the knowledge such as process assessment, measurement, reviews, and testing. The latter is for ML-based systems, because engineers have poor knowledge how to develop ML-based systems as they are automatically generated, nonlinear and too complex. Traditional process assessment, measurement and reviews are hence ineffective. Frequent, Entire, and Exhaustive Testing (FEET) still works. Engineers have to adopt both the inductive development style for the core ML models and the deductive development style for entire ML-based system.

These guidelines extract five aspects of quality evaluation for ML-based systems: Data Integrity, Model Robustness, System Quality, Process Agility, and Customer Expectation.

Data Integrity relates to the quality assurance of samples of inputs and outputs. This guideline has 11 general checkpoints for statistical considerations, privacy, intellectual property rights, online learning, and quality of the data generator, such as volume and cost, meaningfulness and requirements, relationships between population and sample, bias and contamination, complexity, multicollinearity, outliers and missing values, privacy and confidentiality, intellectual property rights, independence of validation data, and effect of online learning.

Model Robustness relates to the quality assurance of a model generated automatically. This guideline has 11 general checkpoints for the characteristics of neural networks, model performance, generalization, noise, local optima, architecture, hyper parameters, cross validation, data diversity, and degradation.

System Quality relates to the quality assurance of the whole system. This guideline has eight general checkpoints for system-level quality including system performance, validation scope, criticality and frequency of accidents, controllability of the system in accidents, functional safety, security, contribution and localizability of ML components, and explainability and assurability.

Process Agility relates to the quality assurance from the viewpoint of development process. This guideline has 11 general checkpoints for quickness of exploration including short iterations and immediate feedback, scalability, automatability, FEET, appropriate skills and deep understanding, and teamwork.

Customer Expectation relates to the quality assurance for various stakeholders, who may be unfamiliar with ML-based system development. This guideline has eight general checkpoints for extravagant expectation for AI, acceptance of probabilistic behaviour, severity of expectation, optimism for huge data, ambiguity of requirements, compliance, linear and deterministic thinking, and bureaucracy. This axis is the baseline for the other. The higher Customer Expectation is, the higher the other axes need to be.

[1] http://www.qa4ai.jp

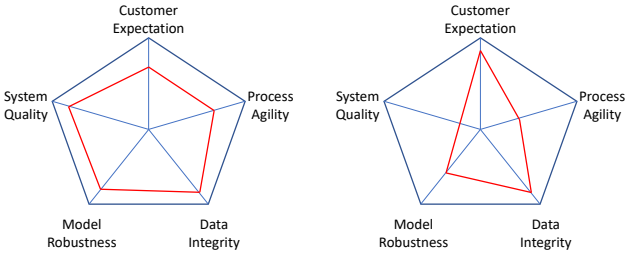

Fig. 1. Examples of Well-balanced and Ill-balanced Quality Pentagon

The total quality of ML systems should be evaluated from the viewpoint of balance among the axes according to Customer Expectation. The development organization of ML-based systems should also establish a well-balanced quality assurance fabric, an organization structure, and a quality management system. Fig. 1 shows examples of a well-balanced and an ill-balanced quality pentagon, consisting of the axes. Furthermore, the total quality of ML-systems usually depends on development phases such as Proof of Concept, Beta Release and deployment of service to a large number of users. The later the phase of development, the better the quality should be.

B. Technical Catalogue

Typically, technical guidelines generalize and summarize techniques and practices being successfully employed in the industry, at least in leading companies. However, for the quality assurance of ML models or ML-based systems, techniques or practices are only just emerging and remain under active investigation. We therefore collected trends from state-of-the-art research papers in the Software Engineering community. We also listed the standard concepts established in the ML community, primarily for performance evaluation, e.g., precision/recall, over/under-fitting, and cross validation.

The state-of-the-art trends we included in the first version of the guideline are as follows.

- Use of pseudo oracle, e.g., [3]
- Metamorphic testing, e.g., [4], [5]
- Robustness evaluation and search for adversarial examples, e.g., [4], [6]
- Structural coverage for neural network [3], [7]
- Methods for explainable AI including local explanation for each output, e.g., [8], [9] and global explanation of the trained model, e.g., [10].

Noted that we endeavor to generalize the concepts as well as decompose multiple aspects combined in one research paper or tool, e.g., in [3].

IV. Guidelines for Specific Domains

The five axes provide common guidelines for the quality assurance of ML-based systems, but it is necessary to design a concrete scheme of quality assurance with an appropriate understanding of the characteristics of each system. Therefore, we examined four popular domains in which ML-based systems are used to discuss the characteristics required quality, and the quality assurance viewpoint for each domain.

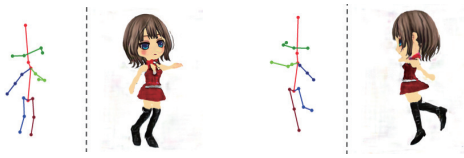

Fig. 2. Image generation from given pose specification.

A. Generative Systems

There have been outstanding advances in techniques for generative models, which learn "what happens with what probability", particularly in techniques for generative adversarial networks (GANs) [11]. With these techniques, applications that create images, videos, essays, or dialogue can be constructed. We focus on such emerging applications because they have a unique focus when it comes to quality: for example, how natural and diverse the outputs are. Such quality attributes are intrinsically fuzzy and difficult to assess automatically.

Our objective in this domain is to uncover potential approaches to automated evaluation of such quality attributes for emerging generative systems. We defined a concrete application that generates an image or video of an anime character, which is inspired by the technique in [12]. Such functions help create attractive interface agents and videos. We defined five use cases for this application. Two of them are shown below and Fig. 2 illustrates the first example.

1) Generate diverse natural character images of a specified pose given as 2D-coordinates of key body parts
2) Generate a natural character video given two images for the start and end points

For these use cases, we enumerated the quality attributes that should be investigated, which are summarized as follows.

- **Naturalness**, e.g., the outputs let human users feel they are created by human creators.
- **Clearness and Smoothness**, e.g., there is no noise, collapse, or discontinuity in the outputs.
- **Diversity**, e.g., poses (when not specified) or clothing in the outputs have a certain degree of diversity.
- **Social Appropriateness**, e.g., no discriminatory or obscene output is generated.
- **Specification Conformance**, the output follows the given instruction such as genders or color of cloths.

Although they are fuzzy intrinsically due to human perception, the possibilities of automated evaluation should be explored. Three primary approaches for evaluating these quality attributes and some of the examples are shown below:

Approach 1 - Metrics: Define and use metrics that represent the target quality attribute, even approximately. For example, we can leverage the evaluation metrics of GANs for naturalness and diversity [13], [14]. As another example, we can evaluate statistical values and distributions of optical flow, which capture the movement of each part in the frames of the video to detect obviously too drastic movement.

Approach 2 - Evaluation AI: Construct an AI that evaluates the target quality attribute. Pose estimation techniques [15]

can be used to judge whether a generated output matches the specified pose. We can also build our own model for pose estimation, as the training data for the generative model originally includes mappings between poses and images, which can be used as training data for a pose estimation model. We can also investigate a dedicated model and data, for the target quality attributes. For example, we may construct a classifier to detect noisy images by creating training data that includes images with noises automatically added.

Approach 3 - Evaluation Rules: Construct a rule-based AI or traditional software to evaluate the target quality attribute. For example, we can implement an analyzer that checks if the specified clothing color is dominant inside the character in the output image.

B. Operational Data in Process Systems

In industrial systems, ML technologies have been applied and practically used in various fields, such as abnormality detection, parameter recommendation and visual inspection. Quality assurance requires the following three characteristics.

- Stakeholder Diversity: Industrial systems consist of multiple subsystems. Data integrity depends on various stakeholders, operations , and contracts.
- Environmental Dependency: Systems are exposed to unrepeatable and unpredictable changes of 5M + E (man, machine, method, material, measure, and environment).
- Accountability: To operate the whole system, we need to endorse the validity for all of system standards and rules.

Considering the three characteristics and the inductive manner in building machine learning model, we defined a development process model for machine learning system, named Intelligent eXperimental Integration (IXI) model as shown in Fig. 3. This model is divided into three phases: proof of concept (PoC), development, and operation. Major risks should be identified and verified in the PoC phase. In the development phase, industrial systems using machine learning are developed, based on the results of the PoC. In operation phase, the output of the deployed machine learning and the behaviour of the system are monitored and maintain its own quality. The results in each phase are collected to explain to stakeholders, and the machine learning model of the system should be evaluated using risk identified data and during operation and would be updated as necessary. The reason why it is difficult to proceed each phase is there are no rational guideline of evaluation in each activities. So we modeled all of mandatory development and operation activities in IXI model, and defined evaluation viewpoints with relationship of quality model below.

- Customer Expectation: Coordinate intangible assets such as software and involved various stakeholders.
- Data Integrity: Repetitive data confirmation process for environmental changes or deterioration of facilities.
- Model Robustness: Condition of data collection and evaluation process and measurements.

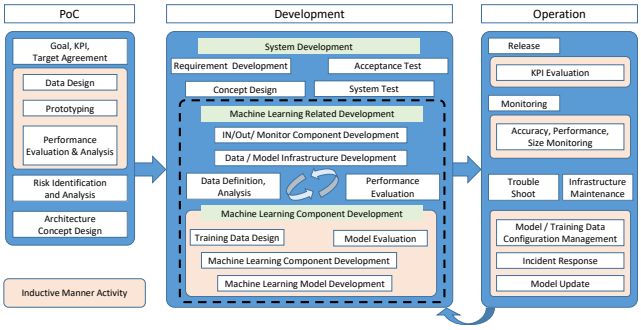

Fig. 3. IXI model : Intelligent eXperimental Integration model

- System Quality: System quality would depend on data and model quality. This criteria shows the evaluation process of each change and explanation to stakeholders.
- Process Agility: Because of above criteria, we emphasized the importance of adapt any changes. We picked up important agile practices.

We also discussed about a real example and added the results to guideline content. It is a system of built-in machine learning system in an industrial machine [16].The system has the three characteristics, so we take it as an good example for our guideline (quality model criteria and IXI model). We discussed criteria, review process and test viewpoints for the case. According to the discussion, we found that there are following pros(+) /cons(-) in our guideline.

- (+) easy to cover quality criteria. It covers all of ML related review points and test viewpoints.
- (+) easy to plan using the IXI phase model. The model helps to understand the necessity of iteration.
- (-) To conclude specific criteria of thresholds and methodology to measure the metrics, still we need ML expert in the project.

Finally, we conclude that our guideline has benefit to cover and plan the quality assurance for industrial system.

C. Voice User Interface System

The voice user interface (VUI) system such as a smart speaker recognizes the user's voice sentence, understands the intent and performs the actions as requested by using the ML technologies as follows.

- Speech recognition: Converts speech signals captured with a microphone into texts
- Natural language understanding: Interprets the converted texts to generate the commands to act
- speech synthesis: Converts texts that are results of the commands to speech signals

We discussed quality of VUI according to the axes of quality evaluation shown in Section III-A. For "Data Integrity", the system requires to perform the same action for the same intention with different voices or expressions. For "Model Robustness", quality of model update is typically important since even new words are created day by day. For "System

TABLE I
EXAMPLE OF TEST ARCHITECTURE FOR VUI SYSTEMS

Test Level	Test Target	Test Viewpoint
Unit test	System modules other than ML	Unit test for each module
	Speech recognition, Natural language understanding and Speech synthesis	Accuracy test for data and ML models
Integration test	APIs	Functional test of integrated modules
System test	Features	Specification-based testing
		Exploratory testing
		Scenario-based testing

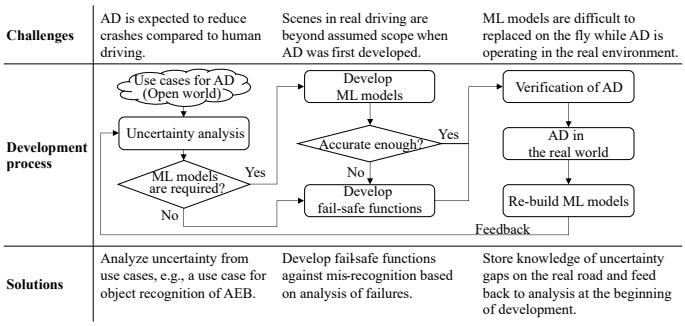

Fig. 4. Methodology for analysis of uncertainty and items to verified in AD development process.

Quality", profiles and daily lives of users are of importance because smart speakers are usually placed home. For "Customer Expectation", it is necessary to determine target users for each function and to evaluate whether the users are satisfied.

The test architecture for the smart speakers consists of several test viewpoints in several test levels as shown in Table I. It is, however, difficult to clearly evaluate the conformity to the requirements due to various requirements for VUIs. The n-level evaluation method will solve such difficulty: Various engineers evaluate whether output behaviors are suitable to various intentions and specifications. An example of five-levels evaluation for the smart speaker is shown below:

1) Perform unintended and different function.
2) The intended function is performed, but the content is unintended.
3) The intended function is performed, but unintended information is returned.
4) The intended function is performed and the intended content is returned, but it must be incorrect.
5) The intended function is performed and the intended content is returned.

Quality assurance levels of the whole system of smart speaker can be defined in the following two levels:

1) Behavior level: The results of tests that can be answered with Yes/No meet the specified acceptance criteria
2) Contents level: The results of tests that evaluate attractiveness of the product meet defined acceptance criteria

D. Autonomous Driving

Autonomous driving (AD) utilizes ML-based systems as core technologies for object recognition, path planning, and manipulation decisions. We investigated ideas, approaches, technologies and methodologies that assure the quality of the ML-based systems, focusing on object recognition for Autonomous Emergency Braking (AEB) as a concrete function of AD and scenario of AEB for the first version of our guidelines. It supports automated steering and acceleration capabilities, which correspond to level 2 of the Society of Automotive Engineers (SAE) standard [17].

We Identified three challenges with the quality assurance of AD: AD is expected to reduce crashes compared to human driving, scenes in real driving are now beyond the scope when AD was first developed, and ML models are difficult to replace on the fly after AD is deployed to real driving. As a solution to these challenges, we developed a methodology consisting of the following phases:

1) Analyze a use-case for object recognition of AEB based on a framework to manage uncertainty for AD [18] and structuring-validation [19]
2) Develop fail-safe functions against mis-recognition based on analysis of failures
3) Store knowledge of uncertainty gaps on the real road and feed it back to the analysis at the beginning of the development

An example of this methodology that includes an AD development process, analysis of uncertainty, and items to be verified in the development process is shown in Fig. 4.

This methodology helps to create test cases for the AEB. For the results of analyzing uncertainty of AEB for pedestrians, test cases are represented as a pedestrian who does not look like a pedestrian (false negative) and an object that looks like a pedestrian (false positive). The false negatives include, for example, pedestrians who wear a coat the same color as the wall or who stand behind a pole, and the false positives include a pedestrian reflected in a window and a painting that looks like a pedestrian. The test cases require expected results. A false negative means the AEB will not work, so the driver needs to operate the brake. A false positive means that AEB will work (the car will decelerate) against the driver's expectation, so the driver cannot avoid the deceleration.

V. EVALUATION

We administered a questionnaire survey to evaluate usefulness of guidelines. The respondents were 31 of the readers, including 13 persons had participated in developing the guidelines, since the authors are also users of the guidelines. Table II shows the professions of respondents.

The questionnaire utilized 5-point Likert scale ranging from "strongly deny" to "strongly agree." The summary of questions is listed in Fig. 5. The result of question 1 shows that the whole of users can understand the characteristics of ML-based

TABLE II
PROFESSIONS OF RESPONDENTS

Target	Research	Devel.	Testing	Quality	Other	Total
ML/AI	1	11	3	2	0	17
Software	1	0	4	8	0	13
Procurement	0	0	0	0	1	1
Total	2	11	7	10	1	31

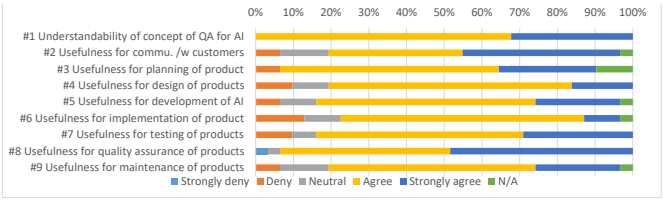

Fig. 5. Responses to questionnaires

products that completely differs from that for software and the proposed concept of quality assurance of them.

Questions from 2 to 9 address the usefulness of the guidelines at each phase of AI-based system development. Over 77% of respondents agreed or strongly agreed with the usefulness at every phase, especially 94% of them did at the quality assurance phase. These results mean that the QA4AI guideline meets the objective of clarifying general concept of quality assurance of AI-based systems.

VI. Threats to Validity

There were few respondents to the questionnaire, so more readers are needed to properly evaluate the guidelines. Moreover, because this was an open web questionnaire, it is possible that only readers who felt positively responded.

Fig. 6 shows the difference of the response to question 1 between the authors and the others. The authors rate the understandability of the guidelines more highly than the others The authors may have a weaker assessment of the guidelines than the others, otherwise, the results may indicate the effect of the consortium's deeper understanding of machine learning properties as they were involved in the development of the guidelines.

VII. Related Work

Reports on practices or case studies are emerging from the industry. Most are general, such as [20], [21], and aspects of quality assurance or testing are very limited. Simple questions to evaluate testing activities were provided in [22]. These questions provide significant guidance on which aspects should be considered, e.g., monitoring input features. Our guidelines, which cover these questions, provide more detailed guidance including the investigation of specific domains in depth.

VIII. Concluding Remarks

We have reported the active efforts for the quality assurance of ML models and ML-based systems in the QA4AI Consortium driven by the Japanese industry. The first version of a set of guidelines was published, including five axes of evaluation,

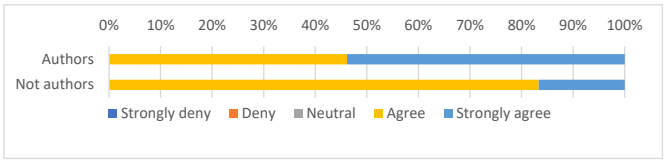

Fig. 6. Differences of responses to question 1 between authors and the others

a technical catalogue, and specific insights for four application domains. Testing is the most significant aspect of the guidelines as testing is the most significant activity in practice. The guidelines provide insights from quality-assurance engineers and test engineers. This direction complements specific testing techniques that have been actively investigated, which are also introduced in the guideline.

Given the high demands of the industry, we opted for a quick release and frequent cycles of updates. We are aware that the current guidelines are insufficient for some aspects of the industry. The first version was constructed in a bottom-up, best-effort way to identify what is missing in the guidelines or in the knowledge from research communities. For example, we found there is very little discussion on how to make use of explainability tools such as LIME [8] in engineering activities.

We are continuously working to extend and enhance the guidelines. Current activities include case studies to uncover more insights in each domain as well as to clarify mapping with other standards such as the Ethics Guidelines in the European Commission[2] and quality standards for general software systems (SQuaRE, ISO/IEC 250XX series).

Acknowledgements

The authors are grateful to all members of the QA4AI Consortium who contributed to the first version of the guidelines. The authors are listed in alphabetical order, with no difference in their contribution to the paper, as representatives of the consortium.

References

[1] F. Ishikawa and N. Yoshioka, "How do engineers perceive difficulties in engineering of machine-learning systems? - questionnaire survey," in *Joint International Workshop on Conducting Empirical Studies in Industry and 6th International Workshop on Software Engineering Research and Industrial Practice (CESSER-IP 2019)*, May 2018.

[2] QA4AI consortium, "Guideline for quality assurance of ai-based products (in japanese)," http://www.qa4ai.jp/QA4AI.Guideline.201905.pdf, Japan, May 2019.

[3] K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore: Automated whitebox testing of deep learning systems," in *The 26th Symposium on Operating Systems Principles (SOSP 2017)*, October 2017, pp. 1–18.

[4] Y. Tian, K. Pei, S. Jana, and B. Ray, "DeepTest: automated testing of deep-neural-network-driven autonomous cars," in *The 40th International Conference on Software Engineering (ICSE 2018)*, May 2018, pp. 303–314.

[2]https://ec.europa.eu/futurium/en/ai-alliance-consultation/guidelines

[5] A. Dwarakanath, M. Ahuja, S. Sikand, R. M. Rao, R. P. J. C. Bose, N. Dubash, and S. Podder, "Identifying implementation bugs in machine learning based image classifiers using metamorphic testing," in *The 27th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2018)*, July 2018, pp. 118–120.

[6] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu, "Safety verification of deep neural networks," in *The 29th International Conference on Computer Aided Verification (CAV 2017)*, July 2017, pp. 3–29.

[7] L. Ma, F. Juefei-Xu, F. Zhang, J. Sun, M. Xue, B. Li, C. Chen, T. Su, L. Li, Y. Liu, J. Zhao, and Y. Wang, "Deepgauge: Multi-granularity testing criteria for deep learning systems," in *The 33rd ACM/IEEE International Conference on Automated Software Engineering (ASE 2018)*, r 2018, pp. 120–131.

[8] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in *The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2016)*, August 2016, pp. 1135–1144.

[9] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *The 34th International Conference on Machine Learning (ICML 2017)*, August 2018, pp. 1885–1894.

[10] E. Angelino, N. Larus-Stone, D. Alabi, M. Seltzer, and C. Rudin, "Learning certifiably optimal rule lists for categorical data," in *The 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2017)*, August 2017, pp. 35–44.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, December 2014, pp. 2672–2680.

[12] K. Hamada, K. Tachibana, T. Li, H. Honda, and Y. Uchida, "Full-body high-resolution anime generation with progressive structure-conditional generative adversarial networks," in *The 1st Workshop on Computer Vision for Fashion, Art and Design*, Septembe 2018.

[13] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *The 29th International Conference on Neural Information Processing Systems (NIPS 2016)*, December 2016, pp. 2234–2242.

[14] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *The 30th International Conference on Neural Information Processing Systems (NIPS 2017)*, December 2017, pp. 6626–6637.

[15] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, June 2019, pp. 2672–2680.

[16] Y. H. Tsuruta Kosuke, Minemoto Toshifumi, "Development of ai technology for machine automation controller (1)," OMRON technics, Tech. Rep., 2018.

[17] S. O.-R. A. D. Committee *et al.*, "Sae j3016. taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," tech. rep., SAE International, Tech. Rep., 2016.

[18] K. Czarnecki and R. Salay, *Towards a Framework to Manage Perceptual Uncertainty for Safe Automated Driving: SAFECOMP 2018 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Västerås, Sweden, September 18, 2018, Proceedings*, 01 2018, pp. 439–445.

[19] L. Gauerhof, P. Munk, and S. Burton, *Structuring Validation Targets of a Machine Learning Function Applied to Automated Driving: 37th International Conference, SAFECOMP 2018, Västerås, Sweden, September 19-21, 2018, Proceedings*, 01 2018, pp. 45–58.

[20] M. Zinkevich, "Rules for reliable machine learning: Best practices for ML engineering," NIPS 2016 Workshop on Reliable Machine Learning in the Wild, December 2017.

[21] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, and T. Zimmermann, "Software engineering for machine learning: A case study," in *The 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP 2019)*, May 2019, pp. 291–300.

[22] E. Breck, S. Cai, E. Nielsen, M. Salib, and D. Sculley, "What's your ML test score? a rubric for ML production systems," NIPS 2016 Workshop on Reliable Machine Learning in the Wild, December 2017.