Quality Assurance for Machine Learning – an approach to function and system safeguarding

Alexander Poth
Volkswagen AG
Wolfsburg, Gernamy
alexander.poth@volkswagen.de

Burkhard Meyer
Audi AG
Ingolstadt, Germany
burkhard.mayer@audi.de

Peter Schlicht
Volkswagen AG
Wolfsburg, Germany
peter.schlicht@volkswagen.de

Andreas Riel
Grenoble Alps University
Grenoble, France
andreas.riel@grenoble-inp.fr

***Abstract*— In an industrial context, high software quality is mandatory in order to avoid costly patching. We present a state of the art analysis of approaches to ensure that a specific Artificial Intelligence (AI) model is ready for release. We analyze the requirements a Machine Learning (ML) system has to fulfill in order to comply with the needs of an automotive OEM. The main implication for projects relying on ML is a holistic assessment of possible quality risks. These risks may stem from implemented ML models and spread into the delivery. We present a methodological quality assurance (QA) approach and its evaluation.**

Keywords*— *artificial intelligence, machine learning, quality management, quality assurance, risk management

I. Introduction

Software engineering as well as corresponding QA and testing approaches have been developed and enhanced over decades. Unlike traditional software, the outcome of a connectionist ML system is highly intractable and non-transparent due to its mathematical complexity, as well as the dependency of its behavior on both the training and in-use data and processes. This gives rise to the necessity of new measures for understanding and explaining ML to the level of rigor required by QA requirements [1]. As a holistic approach ("a system is more than just the assembly of its parts"), a connectionist ML algorithm is locally deterministic, well understood and explainable, however unforeseen behavior may emerge at a global level.

Investigating modern ML development organizations and approaches, we made the following key observations:

- ML is a new and emerging type of software still missing adequate quality measurement metrics, control and assurance techniques [2].
- The general class of software systems with no reliable test oracle available is sometimes known as "non-testable programs". ML-based software belongs to this class of systems.
- Creating mature high-quality products and services is very hard. QA principles should be pro-actively incorporated by design beforehand, instead of reacting on quality claims during product/service use. Quality has to be inherent to the product/service by design instead of being an add-on introduced at later life-cycle stages.

These key observations imply four essential questions to be addressed by a QA approach to ML software:

1. How to identify and estimate quality risks?
2. How to define adequate quality assurance activities to mitigate or reduce quality risks?
3. How to assure being the "right track" to generate customer confidence in the AI/ML software/model at release time?
4. How to deal with non-deterministic behavior of algorithms in QA approaches?

Our systematic methodical approach presented in this article helps to answer these questions systematically. The proposed *evAIa* (evaluate AI approaches) is mainly based on the current state of the art, as well as a mindset to future development in the AI domain especially for its sub-domain ML and their QA and test methods. As currently many industrial products and services are developed with ML-based components, the quality aspect for these new type of data driven functionality and behavior needs adequate safeguarding and QA. The ML-based components can have the objective to add a feature or functionality to a product or service. Furthermore, the ML-based components can be a core part of the product or service offer. In both cases, adequate safeguarding is needed. Depending on the ML-based component, the safeguarding scope has to be on the function or system level. Currently established systematic approaches on industry level are available like the ISO/IEC/IEEE 29119 series for software testing standard, but they are missing a link to AI and ML safeguarding and QA. Our approach to AI QA has been designed to meet the following key requirements:

- Support the life cycle from development to production;
- Fulfill business needs as well as technical aspects;
- Be independent of a particular development or operation process model (in order to assure its applicability in both e.g. V-model and Scrum contexts);
- Be usable based on a hands-on guide by the responsible teams;

978-1-7281-8913-0/20/$31.00 ©2020 IEEE
DOI 10.1109/QRS51102.2020.00016

- Reflect the state-of-the-art;
- Extend easily to new insights or future technologies.

In order to validate the completeness and practical relevance of our approach, we evaluated our questionnaire on a series of case and field studies (section 4) within the Volkswagen Group. The feedback of these AI and ML experts reflects the different working methods and technologies, which are used in the different brands and domains. Currently the approach is offered via the group wide AI working group and their knowledge base. The approach is periodically reflected and iteratively enhanced with the feedback of AI and ML experts.

Section 2 introduces related work, section 3 presents the evAIa method, section 4 evaluates evAIa in the Volkswagen AG and section 5 concludes and give some outlook.

II. RELATED WORK

Recently, research on the software development process and the data processing of AI pipelines has received considerable attention when researchers attempted to understand and improve AI approaches. So far, software quality management is not yet commonly established in this domain. In this section, we present a selection of relevant quality aspects from published work in order to gain an impression of the state of the art in QA and testing in the context of AI approaches. This state of the art will be the base for a generic QA method for AI approaches with focus on ML and their products.

To identify and collect the state of the art, we conducted a literature analysis in the first quarter of 2018 focusing on practical approaches and methods and aligned with [3]. The search was conducted in IEEE Explore, Springer and Elsevier. The used search term was (“AI” or “artificial intelligence” or “ML” or “machine learning) and (“safeguarding” or “quality assurance” or “QA”). We filtered the results by the following criteria:

- Can the content be integrated in a generic QA approach?
- Is the content proven in use?
- Can aspects of the content be rephrased into practical questions or offer a way of measurement or indicator?

The relevant results are consolidated in Table 1 and subsequent tables in chapter 3. To summarize the insights of the literature analysis, we can conclude that for specific aspects of ML and AI quality related approaches exist.However, we could not find any holistic safeguarding approach for ML-based components. Especially the holistic QA in the context of the product or service life-cycle of ML-based components apparently has not been addressed systematically. This result gave the impetus for the development of an approach to address our demand.

TABLE I. QA TOPICS REGARDING AI

Topic	Description/Driver
Test your features and data	
Know your product’s quality risks	Identify quality risks in the product stemming from AI and ML models [4].
Beware your features	ML cannot find gold, where there is none. Adequate feature extraction and/or engineering is important.
Bias of learning data	Identify inherent bias of used learning and training data [5].
Completeness of training data	Identify the boundaries or limitations of the learning and training data and their completeness inside their boundaries. Naturally, these limitations depend on the intended use case and applied technology, but in generally this is one of – if not the – most important challenge.
Design for testability	Validation criteria, methods and procedures for ML models have to be taken into account already during the requirements and design phases of AI-based functions [6].
Test your implementation	
Software code for training and/or serving	Assure the quality of the code of the training and serving pipeline with software engineering QA and testing approaches[4]
ML models used as pre-learned (maybe third-party) library/framework	Get guaranties about the third party’s model quality (define criteria for proven usage in the target domain, etc.) or set up tests to ensure the quality.
White boxing the ML algorithm	Make the way to the outcomes transparent. Search for example for hot spots (single points of failure) in your trained model or test what happens if some nodes are “offline” with the outcome [7].
Test the impact of each tunable hyper parameter	Complex training and service pipelines should have tests about misconfiguration and a configuration management to reproduce the environment and its outcomes [4].
Run multiple versions of models in “diffy” mode	Assure that the same stimulus generates the same outcomes on a serving environment [8][9].
Correct behavior of the data-processing and training pipeline	Correct behavior includes privacy controls across its entire data pipeline for compliance to regulations (i.e. general data protection regulation GDPR).
Risks from importing external ML libraries / frameworks	Get guarantees about external code quality (define criteria for proven in use etc.) or setup tests to assure the quality[4].
Architecture/design guide-lines for ML software	Is the software for the training and testing chain developed according to (business) domain relevant QA guidelines[10]?
Test your infrastructure	
Identify worst cases scenarios in performance to assure required real-time behavior	Check the performance of the trained model about compute, memory, and storage and network bandwidth consumption with “worst-case” stimuli before serving.
Observe output quality during serving	Setup a continuous monitoring of the serving model about the outcome quality.
Define criteria for the verification and validation context in the product domain	Formal evidence about correct engineering is that the state of the art target need to be up to date – this target typically lifts during the time and enforces requalification of newer versions of the model. Define how to observe the “lifting-drivers”.
Identify relevant data protection laws of the target countries and/or users	Assure that the training and serving phase is aligned with the current data privacy and protection laws.

For conventional software, several QA metrics may be applied to evaluate a system. Examples are coverage metrics like C0 (statement coverage) or C1 (branch coverage). However, for AI, these metrics are only of limited use because the model performance is mostly driven by learning data it is difficult to use these metrics or transfer them easily to the ML-based components. Thus this work does not focus on any specific metric, however if a metric is available, it will be useful to measure the complexity and quality of a system. To demonstrate the value of our approach evAIa compared to classic quality assurance approaches, we focused on the topics listed in Table 1.

III. The evAIa Method

A. Context and Overview of evAIa

The evAIa (evaluate AI approaches) method reflects state of the art approaches for QA in AI projects and presents current recommendations to data scientists and engineers for ensuring the capabilities of their AI models and making limitations transparent. EvAIa is based on product risk evaluation, a questionnaire about the used AI-approaches, QA method recommendations to mitigate specific product risks caused by the AI-approach, as well as a transparency report to show the mitigation and residual risk reached with the evAIa approach.

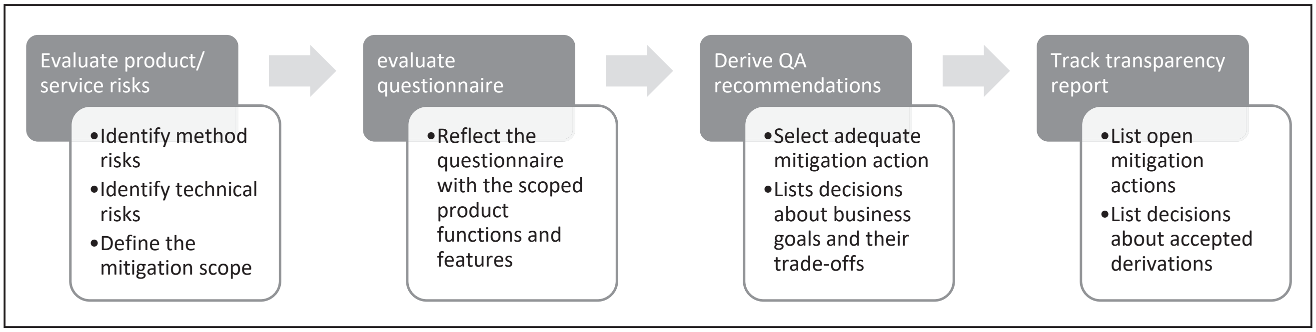

Fig. 1. The application sequence of evAIa

Additionally, the questionnaire is mapped to the ISO/IEC 25010:2011 to show the systematic refinement of the standardized characteristics for the AI and ML domain based products and services as required in formal and regulated environments. The mapping of each question to the related main characteristic of the standard is made in Table 5. As some questions easily can be mapped to more than one characteristic, only the best fitting one has been mapped for simplification. The mapping shows that not all ISO characteristics are addressed by evAIa because security and usability are not specific to ML-based components. However, these topics are also relevant for product and service deliveries. We recommend aligning project work with the applicable established domain standards.

B. Detailed evAIa Method Description

The evAIa method is designed to support quality engineers and developers to realize quality by design in four sequential steps (Figure 1). In the first step, the product/service risks are evaluated against potential quality issues and a mitigation scope is set. In the second step, the product/service team answers the evAIa questionnaire to systematically check weaknesses of the ML learning components of their product or service. In the third step, decisions are made about mitigation actions. These include mitigation by design (quality by design) or specific tests for verification and validation. In the last step, the defined actions are tracked and documented to ensure adequate compliance documentation for the entire product/service life cycle.

C. Risk Evaluation

Product/service teams can make a systematic product quality risk (PQR) evaluation with the PQR method [11] and a more elaborated version for practical use as a workshop kit. Volkswagen offers a workshop kit to all product teams for effective PQR elaboration [12]. With the systematically derived quality risks and their classification, the relevant functions or features of the product or service can be identified for focusing on mitigation of the product/service specific quality risks. The PQR approach is used during the first step of the evAIa sequence (figure 1). The outcome of the PQR analysis is used to focus the questionnaire on the product/service team-specific quality risks (step two in the evAIa sequence). This risk-based QA approach helps to align safeguarding resources with the most relevant quality issues.

D. Questionnaire

The following tables show the questionnaire, which is used to reflect the relevant AI-based functions or features of the product or service. The tables go through the three core AI life-cycle phases of data pre-processing (Table 2), implementing (Table 3), and serving (Table 4). Volkswagen provides to the quality engineers and developers spreadsheets to evaluate, comment and remark each question for an adequate documentation of the ML safeguarding.

The tables are based on literature which has elaborated QA or safeguarding approaches in the ML domain and on the experience of the ML experts who are worked and reviewed the design and development of the evAIa approach. The development was oriented on the design science research approach [13].

TABLE II. Questionnaire for AI based products/services - Test your features and data

Topic	Aspect (indicators)	Questions	
Adequacy	Degree of involvement of validation experts in the design phases of the interacting systems, deployed (GPU-)hardware and software modules; definition of validation criteria and scenarios for AI and ML algorithm requirements; existence of both requirements- (includes stories	1.1	Have algorithms and training- and validation-data been co-designed?
		1.2	Are the requirements to training- and validation-data clearly defined?
		1.3	Which parts of the system shall be

	etc.) and scenario-based (includes use-cases etc.) specification of systems and subsystems [9].		validated against requirements?
		1.4	Which parts of the system shall be validated against scenarios?
Bias	Source of raw data before processing is reliable (no manipulation etc.); relevant bias aspects are identified (culture-bias, locality-bias, social-bias etc.) [4].	1.5	What kind of (inherent) bias does the data have?
		1.6	Why is the data set representative for the ML algorithm (learning and testing)?
Completeness	Use cases define boundaries (check with misuse cases); External drivers about data completeness and boundaries are identified (example: security attack vectors are changing over time); adversarial examples, fitting to the (business) domain [14].	1.7	What boundaries or limitations does the data have?
		1.8	What is the criteria set for an adequate completeness of the data inside the boundaries?
		1.9	Are completeness and boundaries constant over time or can external drivers change them?
Process chain	Bias change/added via processing (labeling-criteria, filtering-rules, concatenation-rules etc.); training and test data are not mixed/enriched or under defined and proven aspect "enriched" (under/overfitting aspects are identified); process chain under configuration management (code, parameters etc. and their processing artefacts input & output data to proof determinism) [15].	1.10	Is the splitting of training- and test-data well chosen?
		1.11	Is the process chain to generate the AI and ML based product/service deterministic and robust?
		1.12	Is the code of process chain to generate the AI and ML based product/service engineered with established QA/testing approaches?
Regulations/ Compliance	Different aspects of regulation are listed (user-based, country-based, usage-based etc.); relevant data protection laws (GDPR etc.) for the aspects are listed; assurance that the training and serving phase is aligned with the current data privacy and protection laws (anonymization, masking, deleting etc.); confirmation that the product/service "is legal" [16].	1.13	What data regulations are set in the target market/countries (and all development and hosting locations)?
		1.14	What is the impact to the data processing flow during the training and serving phase?

TABLE III. QUESTIONNAIRE FOR AI BASED PRODUCTS/SERVICES – TEST YOUR IMPLEMENTATION

Topic	Aspect (indicators)	Questions	
Training and testing chain	Code and configurations are under version control, test-suites for the code are established; architecture and requirements are documented; the established software development process is fulfilled [17].	2.1	Is the software for the training and testing chain developed according to relevant/domain specific QA guidelines?
		2.2	Are the used AI frameworks/libraries, which are implementing the algorithms developed according to relevant/domain specific QA guidelines?
		2.3	Are the code and AI and ML frameworks/libraries with their configuration under version control to reproduce outcomes?
Model transparency	Visualization tools for learning-steps or layers etc. are used; models are checked for "hot-spots" (example: deactivation of high connected nodes and their impact to the output can impact the robustness of the model and its usage context); evaluation for over-/under-fitting [18].	2.4	How much of the model can be "white boxed" to validate it?
		2.5	What are useful checks on a "white boxed" model? How does the model / system react to the white boxing outcome?
		2.6	Which kind of "hot-spots" are acceptable?
Model adequateness	Relevant hyper parameters are identified; value ranges of the relevant hyper parameters are evaluated and tuned to optimize the outcome for the product/service context; the ground truth is identified and evaluated for the product/service context [16].	2.7	Which hyper-parameters are available?
		2.8	Which hyper-parameters are useful in the product/service context?
		2.9	Can the chosen approach describe the ground truth?
Model robustness	Gap between demanded product/service specific aspects and model is identified; methods [8] and [19] for checking robustness are applied; versions are run in parallel (diffy mode).	2.10	Which robustness aspects are product/service relevant?
		2.11	What aspects are factors for model robustness?
		2.12	How can robustness be measured for the product/services?
Model completeness	Established methods of derivation of training- and test-data are used; separation of the data into training and test data is well defined and under configuration management; sufficient completeness of data-sets is checked [20].	2.13	How is the training-data derived?
		2.14	How is the test-data derived?
		2.15	How is it assured and measured that both are sufficiently complete?
Third-party (not product team owned) models / frameworks	Licenses check of the model and framework/library is compliant for productive serving; references to users are checked; in case of open source: the project is "active" on e.g. github and its license fit to product; bugs are fixed fast; transparency of test activities is given.	2.16	How reliable (about the usage domain quality aspects) are third party included AI and ML models and frameworks/libraries?
		2.17	What transparency about their quality exists?
		2.18	What are the quality & license risks for the model?
Model fitting	(Pre-trained) models connected to chains are broken down into "model-units" [21]; each model (-unit) is checked for over-/under-fitting effects; model-chains are integrated and step-wise checked; entire model-	2.19	What impact will over-/under-fitting have? Is this being monitored?
		2.20	How can chains be broken down for model-"unit"-testing?

	chain is checked end-to-end; critical model (-unit) is cross-checked with other model-implementations (example: Keras can use a TensorFlow and Mxnet implementation of a algorithm for cross-checking model behavior) or checked by more simple models to assure not to rely on a special implementation or a side-effect of an implementation bug.	2.21	What other model types or implementations can be used for cross-checking correctness (algorithm diversity)?
		2.22	How can the confidence in the output be measured and improved?
		2.23	What are useful integration steps of models to chains?
		2.24	How do test data and/or scenarios "scale" with the integration of several "model-units"?

TABLE IV. QUESTIONNAIRE FOR AI BASED PRODUCTS/SERVICES – TEST YOUR INFRASTRUCTURE

Topic	Aspect (indicators)	Questions	
Configuration	Artefacts are under version control; deployment is automated; changes are avoided or logged [16].	3.1	How to ensure, that the same hyper-parameters for training are used for implementation?
Execution environment	Fix and variable resource allocation is identified; service level agreements (SLA) for resources are defined; adequate log-levels are set; up/down-scaling is checked; availability checks are in place [22].	3.2	Which levels of availability and scalability are needed?
		3.3	How much logging information of the execution environment is needed (for operating and debugging)?
Monitoring	Monitoring relevant model aspects are identified; thresholds and triggers for each aspect are defined [23]; visualization charts about the model's decision quality is established [system and its drift on new context vs human as basis].	3.4	Which model in/output have to observed/monitored?
		3.5	Is the monitoring up and adequate triggers defined?
		3.6	Is the model decision quality and its trend is transparent?
Worst cases	Worst cases are identified (no late or wrong response); impact of missing resources (like compute power, memory size) is tested; appropriate resource set is defined and allocated for stable serving.	3.7	What worst cases for inference exist?
		3.8	Are the resources for worst cases allocated?
Validation	Test cases in the form of requirements- and/or scenario based validation procedures are specified [14].	3.9	Are all of the above aspects (parameter settings, monitoring, and worst-case behavior) covered by test cases?

TABLE V. MAPPING ISO 25010 CHARACTERISTIC TO EVAIA QUESTIONS

ISO Characteristic	evAIa Question
Functional Suitability	1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.13, 2.15
Performance Efficiency	3.2, 3.7, 3.8
Usability	Not explicitly addressed by evAIa
Compatibility	2.16, 2.17
Reliability	1.9, 1.10, 1.11, 1.12, 1.14, 2.1, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 2.10, 2.11, 2.12, 2.13, 2.14, 3.1, 3.3, 3.4, 3.5, 3.6, 3.9
Secuirty	Not explicitly addressed by evAIa
Maintainability	2.2, 2.3, 2.18, 2.21, 2.23, 2.24
Portability	2.19, 2.20, 2.22

E. QA recommendations

Table 6 correlates with the aspects of Tables 2, 3 and 4 in that it proposed related methods or approaches. Their adequateness depends on the business goals and the desired tradeoff between quality risk mitigation and effort to the benefit of the action. Rather than a rule based approach, [9] consider it more as a practice collection for QA inspiration. Furthermore, a business goal often is not only to mitigate quality risks – a business objective can be to push some specific ISO characteristics of Table 5. Step 3 of the evAIa sequence (figure 1) balances the selection of the safeguarding measures. Depending on the specific safeguarding strategy of the product or service, the measures of table 6 are selected by the business related quality risks and focused characteristics (table 5). For both the tables 2, 3 and 4 provide the most relevant topics and issues that have to be addressed in the safeguarding measures which are selected from table 6. To support the engineering of selected quality characteristics, the corresponding questions should be handled with priority. In any case, any QA measures have to include the verification of behavior in case of incorrect or unacceptable data.

F. Transparency report

Based on the identified risks of the risk evaluation and the selected QA recommendations, the outcome of the evAIa approach is a list of actions that make transparent what kind of quality improvements are possible and/or should be done to have a state of the art AI and ML based product or service. Wherever the state of the art is adequate to the needs, no further quality improvement actions are proposed.

TABLE VI. QA PRACTICES FOR PQR MITIGATION

Aspect	QA methods/approaches
Data quality	Analyze technical (distribution, outliers, noise/confidence, slice it, significance and time stability) and process (checked aspects are visible and consistent over time, proof of hypothesis) aspects of the data set; when a data preparation pipeline is used, check for drops etc., define data owners.
Training adequateness	Analyze correlations (reuse) of training and testing data, analyze the hyper-parameters about relevance and reflect tuning behavior, analyze over/undertraining, insert some noise into the data and check quality of the output for robustness [7], for debugging/interpreting results use linear models as long as possible, define policy layers and do not mix up different aspects, find a tradeoff between specific and generalized features, eliminate unused features (technical debt), eliminate undesirable behavior after measuring it.
Model validation	Analyze validation data for independency of training and testing data, benchmark different model libs/ frameworks, analyze outcome quality for adequateness, start with simple models, enhance them iteratively.
Serving	Analyze prediction quality over time, define re-learning triggers, assure that training and serving environments are comparable, code equality to the training environment is as high as possible, and ensure a minimum/maximum prediction rate for checking the "viability" of the system.

IV. EvAIa In Producti Development

A. Context of the evaluation project

This example applies evAIa to a cloud product/service of the Volkswagen Group IT cloud [24], which uses AI for anomaly detection. We use the questions listed in the introduction to demonstrate the benefit of the systematic PQR analysis and application of the questionnaires to the product team even in cases where the evaluation is done after development start.

1) How can we estimate the quality risks?

Based on the product vision and the product features, the PQR analysis is set up. The outcomes are technical (TPQR) and methodical (MPQR) product quality risks. The following is an extract of the main risks identified by our PQR-analysis:

TPQR 1 – Inadequate implementation: The implementation of machine learning based applications is challenging because it involves various successive transformation processes that have to fit together smoothly. The most risky part is the data-preprocessing step that involves the log categorization and the feature representation. An incorrectly implemented transformation leads to insufficient machine learning models and thus to poor predictive capability.

TPQR 2 – Inadequate deployment: Due to the application's complexity, there is a high risk that the data representations in the training mode and in the predictive mode are not mapped equally. However, in order to deploy machine-learning models in production, it is crucial to preprocess the data in the same way as in the training mode. Otherwise, the model is not able to predict anything, because it does not receive the data in the required format.

MPQR 1 – Inadequate data representation: The usage of log files as an input for machine learning algorithms is challenging, since we have to deal with heterogeneous, unstructured data. Various preprocessing steps are required to transform the raw log data into a numerical representation. However, this complex data structure along with the transformations may bear the risk of being insufficient in terms of error prediction.

MPQR 2 – Inadequate model quality: The quality of machine learning models depends on various hyper-parameters and the algorithms themselves. Hence, it is challenging to select the most suitable model among all combinations. However, there is always a risk that machine-learning algorithms are insufficient to model a problem.

For easier and guided application of the PQR approach in a product setting, it is recommended to have a workshop self-service kit for the product workshop team. The Volkswagen AG uses a four-step design thinking based approach to identify systematically quality risks. The approach is offered as optional support tool to the evAIa approach as self-service kit [12].

2) How to define adequate mitigation activities?

The derived action for TPQR 1 is unit testing, integration testing, system testing, and for TPQR 2 integration testing, system testing. As the service does not require the development of its own (specific) ML algorithms, there is no additional software QA beyond the integration of the selected ML libraries into the product specific application code. This leads to the established software QA actions. For mitigating MPQR1 and MPQR2, different forms of feature representations as well as different machine learning algorithms can be investigated. E.g., some performance indicators, such as recall, F-measure and the area under the Precision-Recall curve (PR-AUC) can be applied.

To help the teams apply the presented tables in a product setting, we recommend providing the questionnaire to the teams as a spreadsheet which they can fill with their notes and indicator evaluations. This is what the Volkswagen AG does as part of the evAIa self-service kit. Furthermore, some specific design decisions and QA actions can be associated with the questions. This kind of documentation helps to ensure documentation for traceability and product compliance aspects if relevant. Additionally, the spreadsheet questionnaire template can offer domain-specific examples about expected outcomes or indicators to questions or give further information and links to related information.

3) How to assure being "on the right track"?

The TPQRs are mitigated by established QA actions for software testing. In the presented case, the best machine learning models of each examined algorithm achieved a PR-AUC score of 0.96 in average in the defined prediction use case with the training data set based on a past event stream. Therefore, we can conclude that the applied QA actions to mitigate the risks for the MPQR's are sufficient for the needed outcome quality of the model. However, in the evaluated context the open point is the change of the environment, which leads to the topic "completeness" of the training data with respect to their "worst cases" and "monitoring" not being addressed adequately in the evaluated version.

B. Goals and strategy of evAIa at Volkswagen

The Volkswagen Group – with its brands like Audi and its legal entities like Carmeq - uses the evAIa method to develop and enhance their connectionist AI and ML based products and services. As evAIa is by design independent of the specific development model, it can be applied in the different environments with their specific development approaches like the specific agile methods adopted by the Volkswagen Group. The users' feedback via the internal quality innovation network (QiNET) [25] enables a continuous discussion and enhancement of the evAIa approach. The goal is to offer to all product and service teams a common state of the art approach for QA and testing of AI models in a self-service offer. Furthermore, a common quality practice is the basis for reusing AI models without heavy redesign and requalification of "product external" AI models.

1) EvAIa applications in different domains

An observation of the teams during the application of the evAIa approach was conducted in a wide range of business areas of the enterprise to get generalization insights. To validate the relevance of evAIa to projects of the Volkswagen AG, we confronted evAIa users with the following questions:

A. What insights does the questionnaire-based evAIa approach deliver to the product teams?

B. How do different application domains use evAIa in their daily work?

C. What is missing to achieve a more effective QA?

We present insights from ten projects / product teams of three legal entities of the Volkswagen Group: The Group Research and engineering entity focus on embedded vehicle AI systems, while Volkswagen Group IT and Audi brand IT emphasizes on business digitalization as AI application domains. The projects have a wide range from focus on autonomous driving assistance systems, after sales use cases to IT internal technical use cases. Objectives of the ML models of our evaluation have a wide range from object recognition on pictures to text analysis in streams. The respective results of the evAIa approach are:

A. All teams argued that they did not systematically address all of the evAIa aspects. Especially teams with few AI senior experts needed assistance. This assistance gap is closed with evAIa for QA aspects. The self-service kit get a high acceptance rate by the teams because they are independent in doing their work by their responsibility without external "supervisor" like from a QA department. The evAIa self-service kit fits with the agile mindset about autonomy and mastery.

B. The company's research teams do not deliver production-ready systems or services and can skip some of the "serving" aspects. However, they have to assure that later on their service design is extensible to fit all serving aspects. Based on the different outcomes the questionnaire is seen more as an inspiration to not forget something relevant or the questionnaire is seen as a part of the product or service documentation to confirm that the released version is safeguarded adequately aligned with state-of-the-art approaches.

C. Feedback about the evAIa questionnaire has led to structural improvements, more precise questions with examples to avoid misunderstanding. This has rendered evAIa useable without trained moderators to support the self-service mindset of autonomous teams. Furthermore evAIa helps to close the gap between the established generic and software code driven QA approaches and the ML specific data driven QA aspects. However, an open point which evAIa cannot address is the inherent lack of transparency of how ML algorithms have learned what they have learned. This is still an open research aspect which is important for some businesses cases which demand to demonstrate in a transparent way the decision finding of the AI bases system.

This project validation and feedback loop checks the feasibility of the application of the evAIa approach and prepares the rollout of evAIa for 2020. Feedbacks and lessons learned of the evaluation leads to some small enhancements and the setup of a self-service kit (a check-list and a how-to). The self-service kit is to ensure a scaling application without experienced moderators for evAIa. The rollout quickly establishes the base for a broad empirical analysis. The important actualization of evAIa by periodic investigation and subsequent integration of the rapidly progressing state-of-the-art will be assured by a dedicated working group. The working group has to reflect the progress in research approaches and methods for safeguarding ML products and services with the objective to transfer and integrate them into the applicable state-of-the-art in enterprise ML development and service delivery. Currently the frequency for the periodical update is annual.

The added value of the establishment of the evAIa method is manifold:

- Teams using ML get guidance for their specific product or service safeguarding (developer view);
- Products and services are transparent safeguarded and the QA is documented (governance view);
- The organization establishes a practice to ensure common safeguarding understanding for ML products and services (QM view);
- The approach to enhancing evAIa is open and transparent to ensure currency in the fast developing ML domain (management / organization development view).

V. Conclusion And Outlook

The presented evAIa evaluation activities have proven that the approach provides added value in practice. The systematic questionnaire reveals aspects that need systematic tracking and mitigation. EvAIa inspires actions and measures to improve AI and connectionist ML models and their training and serving environments. In particular, evAIa leads to transparency about the current state of QA. This leads to active decisions about how much additional qualification of the service is useful. EvAIa got fast acceptance for example in a centralized AI competence center which introduced evAIa as a standard for their project QA. As the ISO 25010 mapping indicates, evAIa mostly contributes to safeguarding on the reliability and functional suitability characteristic.

The presented approach is neither a generic assessment model nor a QA standard for AI based products and service. EvAIa is rather an instrument helping to pave the way to a systematic QA for AI based products and services. EvAIa extends the established QA approaches with AI domain specific aspects. With the self-service offer, the integration into the autonomous agile teams is possible as well as in other development approaches like V-model. Furthermore, there is an option to use the evAIa method by central governance or QA instances to compare different business areas or specific service domains about their established ML safeguarding in the future for organizational wide improvements to reach some baselines in ML QA if there is a demand like for agile transitions it came [26].

The future research and development of evAIa includes extending the questionnaire to better address more non-connectionist ML approaches. Furthermore, an investigation about typical patterns on ML safeguarding shall be derived by collecting results of a wider range of product evaluations based on the structured questionnaires of evAIa.

References

[1] Basu S., Seshia S., Wiens J., Wang L., Puri R., "Autonomous Systems and the Challenges in Verification, Validation, and Test," in IEEE Design&Test: 2017 Design Automation Conference roundtable, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8361983&tag=1, 21 May 2018.

[2] Menzies T., Pecheur C., "Verification and Validation and Artificial Intelligence," Advances in Computers Volume 65, pp. 153-201, 2005.

[3] Webster, J., Watson, R. T. "Analyzing the past to prepare for the future: Writing a literature review," MIS Quarterly, 2002, 26(2):13-23

[4] Breck E., Cai S., Nielsen E., Salib M., Sculley, D., "What's your ML test score? A rubric for ML production systems," in Neural Information Processing Systems (NIPS), 2016.

[5] Torralba A., Efros A., "Unbiased look at dataset bias," in IEEE Conference: Computer Vision and Pattern Recognition (CVPR), 2011.

[6] Riel A., Kreiner C., Macher G., Messnarz R., "Integrated design for tackling safety and security challenges of smart products and digital manufacturing," CIRP Annals - Manufacturing Technology, p. 177–180, 04 2017.

[7] Sculley D., "TensorFlow Debugger: Debugging Dataflow Graphs for Machine Learning," in Neural Information Processing Systems (NIPS), 2016.

[8] Zheng S., Song Y., Leung T., Goodfellow I., "Improving the Robustness of Deep Neural Networks via Stability Training," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[9] Evtimov I., Eykholt K., Fernandes E., Kohno T., Li B., Prakash A., Rahmati A., Song D., "Robust Physical-World Attacks on Deep Learning Models," 2018. [Online]. Available: https://iotsecurity.eecs.umich.edu/#roadsigns.

[10] Zinkevich M., "Rules for Reliable Machine Learning," in Neural Information Processing Systems (NIPS), 2016.

[11] Poth A., Sunyaev A., "Effective Quality Management: Risk- and Value-based Software Quality Management," IEEE Software Volume 31 Issue No.6, pp. 79-85, 2014.

[12] Poth A., Riel, A, "Quality Requirements Elicitation by Ideation of Product Quality Risks with Design Thinking," IEEE Conference on Requiremetns Engineering, 2020 (in print).

[13] Winter, R., "Design Science Research in Europe." Euro. J. Inform. Syst., vol. 17, pp. 470-475, 2008.

[14] Kurakin, Alexey, Ian Goodfellow, and Samy Bengio. "Adversarial machine learning at scale." arXiv preprint arXiv:1611.01236 (2016).

[15] Schelter, S., Biessmann, F., Januschowski, T., Salinas, D., Seufert, S., Szarvas, G., Deshpande, A. "On Challenges in Machine Learning Model Management." IEEE Data Eng. Bull., 41(4), 5-15. 2018.

[16] Murphy, C, Kaiser, G. E., Arias, M. "An approach to software testing of machine learning applications." (2007).

[17] Breck, E., Polyzotis, N., Roy, S., Whang, S. E., Zinkevich, M. "Data Infrastructure for Machine Learning." In SysML Conference. 2018.

[18] Cai, S., Breck, E., Nielsen, E., Salib, M., & Sculley, D. "Tensorflow debugger: Debugging dataflow graphs for machine learning." 2016.

[19] Mansour Y., "Robust Learning and Inference," in Neural Information Processing Systems (NIPS), 2016.

[20] Provost, Foster. "Machine learning from imbalanced data sets 101." In Proceedings of the AAAI'2000 workshop on imbalanced data sets, vol. 68, no. 2000, pp. 1-3. AAAI Press, 2000.

[21] Dahl, G.E., Yu, D., Deng, L., Acero, A., "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition." IEEE Transactions on audio, speech, and language processing, 20(1), pp.30-42. 2011.

[22] Ishakian, V., Muthusamy, V., Slominski, A., "Serving deep learning models in a serverless platform." In 2018 IEEE International Conference on Cloud Engineering (IC2E) (pp. 257-262). IEEE. 2018.

[23] Lin, J., Kolcz, A., "Large-scale machine learning at twitter." In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data (pp. 793-804). 2012.

[24] Poth A., Werner M., Lei X., "How to Deliver Faster with CI/CD Integrated Testing Services?," In: Larrucea X., Santamaria I., O'Connor R., Messnarz R. (eds) Systems, Software and Services Process Improvement. EuroSPI 2018. Communications in Computer and Information Science, vol 896 pp. 401-409, Springer, Cham, 2018.

[25] Poth A., Heimann C., "How to Innovate Software Quality Assurance and Testing in Large Enterprises?". In: Larrucea X., Santamaria I., O'Connor R., Messnarz R. (eds) Systems, Software and Services Process Improvement. EuroSPI 2018. Communications in Computer and Information Science, vol 896. pp 437-442 Springer, Cham, 2018.

[26] Poth A., Kottke M., Riel A., "Scaling Agile – A Large Enterprise View on Delivering and Ensuring Sustainable Transitions." In: Przybyłek A., Morales-Trujillo M. (eds) Advances in Agile and User-Centred Software Engineering. LASD 2019, MIDI 2019. Lecture Notes in Business Information Processing, vol 376, pp. 1-18. Springer, Cham, 2020.