World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:15, No:3, 2021

Machine Learning Development Audit Framework: Assessment and Inspection of Risk and Quality of Data, Model and Development Process

Jan Stodt, Christoph Reich

***Abstract*—The usage of machine learning models for prediction is growing rapidly and proof that the intended requirements are met is essential. Audits are a proven method to determine whether requirements or guidelines are met. However, machine learning models have intrinsic characteristics, such as the quality of training data, that make it difficult to demonstrate the required behavior and make audits more challenging. This paper describes an ML audit framework that evaluates and reviews the risks of machine learning applications, the quality of the training data, and the machine learning model. We evaluate and demonstrate the functionality of the proposed framework by auditing an steel plate fault prediction model.**

***Keywords*—Audit, machine learning, assessment, metrics**.

I. Introduction

MACHINE LEARNING (ML) and Artificial Intelligence (AI) are increasingly used in many sectors of the economy. Especially in areas such as automation, sensor technology, assistance systems, predictive maintenance and resource management. AI-based applications have been implemented particularly successfully in these areas [1]. The technology thus makes a decisive contribution to securing or improving the respective market position. However, this is accompanied by various risks based on ”black box” modeling as a basic principle of AI. Conventional computer programs process data by explicit instructions or commands defined by software to solve a problem. In contrast, AI is based on independent learning processes, creating system autonomy, which leads to completely new approaches to problem solving. The complexity of machine learning algorithms used makes it difficult or currently impossible for data scientists to follow the decisions made by the machine learning algorithm.

Therefore, there is no guarantee that an AI application will always reliably deliver good results. However, this is a requirement that must be met by an autonomous vehicle, for example. A common question that arises in the area of critical AI applications is how to prove safety while at the same time not knowing how the software will behave.

First of all, you need to know the risk, secondly you need to know what the requirements are for your AI-based application, and thirdly you need to have a development process that guarantees that the requirements are met. In order to achieve high quality of your AI application, so that the customer keeps a high level of trust in the product and legal problems are avoided, an audit framework is required to ensure that the AI application meets the requirements.

This paper is structured as follows: Section II gives an overview of the related work. Then the paper refers to the ML-specific audit III and refines the ML assessment of data, ML model and development process in IV. In Section V an audit framework is introduced and evaluated in Section VI. Finally, a conclusion is drawn in Section VII.

II. Related Work

This section has been divided in a) ML testing frameworks for predictive models, b) metrics for data quality, c) metrics for ML models d) data and ML model fairness:

a) There is little work on testing predictive models. Most of them focus on the quality assessment of predictive models, which is an important part of an audit, but not all of it. An overview on benchmarking machine learning devices and software frameworks can be found in Wei et. al [2]. Nishi et al. [3] developed a quality assurance framework for machine learning products and their underlining model. The framework consists of a set of metrics that define model quality, a model evaluation strategy, and development lifecycle testing to maintain model quality. Bhatt et al. [4] point out that an audit framework should include evaluation of the conceptual soundness of the model, monitoring, and benchmarking of the model, and should provide result analysis. Zhang et al. [5] give a comprehensive overview of the state of the art in the field of machine learning testing. Workflows to be tested, metrics and characteristics of machine learning are presented.

b) Auditing machine learning models requires a known quality of the test data. Stewart et al. [6] show in detail the impact of poor data quality on different machine learning algorithms and their accuracy and performance. Schelter et al. [7] provide an overview of process, models and metrics for validating the quality of labeled data against application-specific requirements. Barrett et al. [8] propose

Jan Stodt is with the Institute of Data Science, Cloud Computing and IT Security, Hochschule Furtwangen of Applied Science, 78120 Furtwangen, Germany (phone: +497723920-2379; e-mail: jan.stodt@hs-furtwangen.de).

Christoph Reich is with the Institute of Data Science, Cloud Computing and IT Security, Hochschule Furtwangen of Applied Science, 78120 Furtwangen, Germany (e-mail: christoph.reich@hs-furtwangen.de).

This work has received funding from INTERREG Upper Rhine (European Regional Development Fund) and the Ministries for Research of Baden-Wuerttemberg, RheinlandPfalz and from the Region Grand Est in the framework of the Science Offensive Upper Rhine for the project HALFBACK. The authors would like to thank the Master student Janik Baur for doing programming and evaluation support.

methods and metrics to evaluate and improve the quality of hand labeled data using statistical approaches.

c) Regarding the metrics for evaluating the quality of machine learning, Handelman et al. [9] provide an overview and evaluation of metrics of machine learning models for gaining insight into the machine learning model. Noise sensitivity [10] or robustness [11], [12] are important metrics in an audit.

d) Another recent research area related to audits is the fairness of machine learning algorithms. Rick et. al [13] investigated how well different forms of audit are suitable for this purpose. Not only predictive models were considered, but potentially all algorithms for decision making. The object of an audit would be understood as a black box. In order to achieve transparency, works of Waltl et. al [14] or Arrieta et. al [15] where used, which try to explain the mechanisms of the neural network within the black box.

III. Machine Learning Specific Audit

The audit process includes the following steps: a) Planning, b) Definition of audit objectives and scope, c) Collection and evaluation of evidence, d) Documentation and reporting. Is necessary to understand that there is a trade-off between costs and risks that management must accept. The nature of ML-based applications differs from traditional software in several features. Before a risk assessment for the audit objectives and scope can be performed, some general objectives must be described, which are determined by the nature of ML-based applications.

A. General Objectives of Machine Learning Audit

Like any algorithm- and data-driven process, ML gives the internal audit a clear role in ensuring accuracy and reliability. ML can only function properly if it analyzes good data and evaluates it against valid criteria - areas where internal audit can have a positive impact. An audit is a formal review of an item or process with the objective of examining the enforcement of policies and guidelines to mitigate risk (see next section III-B). The nature of ML-based applications imposes some additional objectives:

- ML applications should be classified into different risk classes. Depending on the risk, they can then be approved, reviewed or even continuously monitored.
- The testing of ML systems with high or very high risk should be carried out by independent testing organizations. The risk-based approach is an established procedure of the European Single Market to combine security and innovation.
- A prerequisite for the manufacturer-independent testing of algorithmic systems is access to the safety-relevant data required for this purpose (e.g. data for driver assistance systems in cars).
- Continuous verification is necessary for the learning of ML systems, since variations of the data (newly collected data) lead to new models.
- Besides the verification of ML models, verification data is essential.

B. Risk Assessment for Audit Objectives and Scope for Machine Learning based Application

There are ML opportunities and risks [16], which can be divided into economic risks, such as the acceptance of AI-based applications by the client, etc., and technical risks. Technical risks are of utmost interest to be considered in an audit process.

- *Logical Error:* Like any software, ML is subject to the risk of logic and implementation errors. This can affect the effectiveness of algorithms, which can lead to a reduced quality of results and thus to massive impacts on the application context.
- *Human Factor and Biases:* There is a risk that unintentional human bias may be introduced into the design of the AI. Due to the lack of knowledge of a domain, data for the training of neural networks might be missing, which reduces the quality of the result. The results of neural networks are interpreted by humans and should not be taken for granted (e.g. in cancer diagnosis).
- *Data Quality:* The quality of ML results depends on the input data, therefore the input data quality is crucial. Achieving data quality involves checking for consistency, balance, robustness, accuracy, compatibility, completeness, timeliness, and duplicate or corrupted data sets. The training data must be representative of the real data, in particular it must be ensured that the sampling rate is appropriate. It must also be considered that the data sets are noisy, have outliers, missing values and duplicates.
  For example, robustness means safe behavior despite unexpected or abnormal input data [17]. It should be ensured that the intelligent system containing an ML model is safe to operate with acceptable risk. If the model, in the example in Fig. 1, receives an unexpected image (e.g., darker image) instead of the trained image of a traffic situation, it must not attempt to recognize it as a new traffic situation. The operating limits defined by the data set used for the training must be taken into account.

Fig. 1 Error of Autonomous Vehicle, because of Brightness Changes [18]

- *Model Quality:* Modeling quality describes how well an ML model reflects reality and thus fulfills its purpose. It

considers how accurate the predictions of a predictive model are. When testing the modeling quality, the predictions of the predictive model for several test data are compared with the correct values, the ground truth [14]. As Fig. 2 shows, the comparison of the predictions ŷ and the ground truth is y*. Ideally the result of the black box is identical to the ground truth. ML model metrics try to quantify the expected difference to the ground truth. The model must be correct (correct ML type and architecture) - otherwise it will never fit the data well, no matter how long the training phase will be or how good the data might be.

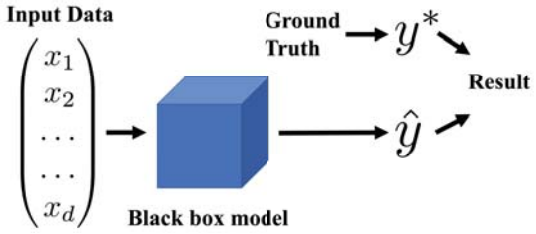

Fig. 2 Black Box Model Predicting the Ground Truth

- *Cyber Security:* Digital attackers get an additional attack vector by using AI. The integrity of ML models could be compromised, e.g., noise is added to an X-ray image, thereby turning predictions from normal scan to abnormal [19], or confidentiality could be compromised, e.g., private training data used to train the algorithm has been restored and faces have been reconstructed [20]. Furthermore, the type of information processed (personal / sensitive company data) can be an additional motivation for attacks.
- *Compliance:* National and international legislation is not (yet) fully adapted to the use of AI. This leads to partially unclear legal situations and legal risks for the involved actors. At the same time, there are strict regulations for parts of the AI, e.g. in connection with the mass processing of personal data by the European Data Protection Basic Regulation (DSGVO). Non-compliance with the requirements in this sensitive area is sanctioned with heavy fines.

IV. Machine Learning Assessment: Data, Model, Development Process

Data, ML models and the ML development process are most important to be reviewed and therefore investigated in detail.

A. Data and ML-Model Inspection With Metrics

Fig. 3 shows a selection of metrics for regression and classifier quality assessment metrics. This choice of metric has an impact on the test result, since the different metrics focus on different quality criteria. For example, both the MSE and MMRE metrics calculate the deviation from the prediction to the ground truth, but use different formulas. MSE calculates the mean squared error, while MMRE calculates the mean relative error with respect to the correct value. There are also compound metrics that try to combine several basic metrics. It is possible that predictive models perform well with one metric but worse with another [21]. Which metric is most appropriate can only be determined in relation to the specific application.

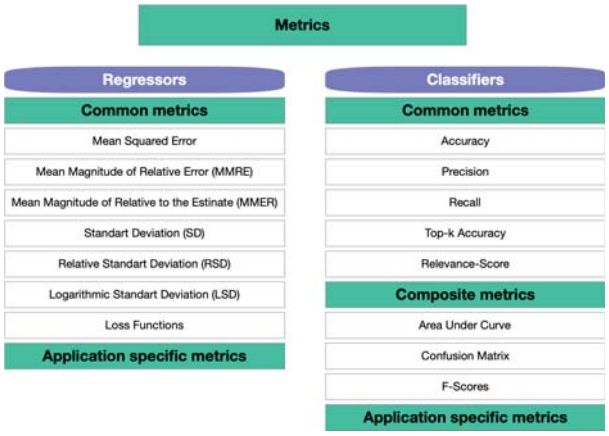

Fig. 3 Selection of Quality Metrics

B. ML Development Process Inspection

ML-engineering and -development is today carried out and used in almost every organization to a different extent - quite analogous to the general software development over the years. However, ML software has unique features that clearly distinguish it from traditional enterprise software development. The ML development process to be investigated can be described as shown in Fig. 4, the ML development process under investigation can be divided into 3 tasks: the validation of data, the validation of ML models, and the validation of the ML application itself.

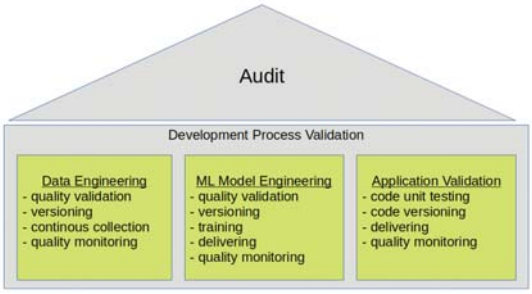

Fig. 4 Assessment Overview

The details to be considered are:

- *Continuous Data Quality Check:* The models are based on historical data, while the data itself is constantly changing. Data quality is often taken for granted without having been properly tested and validated, and its quality must be continuously reviewed. Therefore, data quality must be continuously checked, data must be versioned, continuously collected to represent the ground truth and

correspond as closely as possible to the real world, and data quality must be continuously monitored.

- *Continuous Model Quality Check:* ML models must be subject to continuous quality checks. Versioning, model provisioning and quality monitoring of the processes must also be ensured. And regular model updates through training and ensuring the same quality are important.
- *ML DevOps - Continuous Delivery:* The quality of traditional software development must be ensured. In addition, however, the integration effort of ML models into the product must also be regulated. How quickly changes to the ML models developed by data scientists can be integrated into the product lifecycle is crucial for the timeliness of an ML application. ML-DevOps must therefore be supported.

Therefore a special ML software development lifecycle (ML-SDLC) with its quality indicators should be reviewed by the auditors.

V. Machine Learning Audit Framework

First the specification of audits is discussed, followed by the ML audit framework, its workflow description, and possible implementation.

A. Machine Learning Audit Specification

To the best of our knowledge, there are currently no machine-readable specification languages for machine learning (see [22]). For continuous auditing and automation an xml-based audit language for ML is required. An overview and comparison of audit specification languages in general and specifically for cloud computing was presented by Doelitzscher et al. [23]. Based on the overview in relation to the domain of machine learning, there are the data and ML models that need to be quality checked by a number of metrics, such as accuracy, completeness, timeliness, etc. The ML development process is a manual task by the internal or external auditor.

B. Machine Learning Audit Framework Architecture

The framework architecture consists of four modules, which in turn consist of several sub-modules. Fig. 5 visualizes the degree of abstraction of the modules by the vertical arrangement. The topmost module, *Audit*, has the highest degree of abstraction. This is where the administrative and organizational processes take place, such as the audit process, the definition of the goals of an audit. The *Inspection* module covers the targeted activities of an auditor to demonstrate compliance with guidelines or specifications. *Resources* are entities that are required to perform an audit. The *Toolbox* module does not include entities, but rather the tools and actions from which an audit can be assembled.

In summary, the Audit module specifies the audit objectives according to the assessment requirements of the objective, the ML application. The more detailed circumstances of the audit that affect the modules Audit and Resources are defined and provide evidence to the auditor. The specifications determine

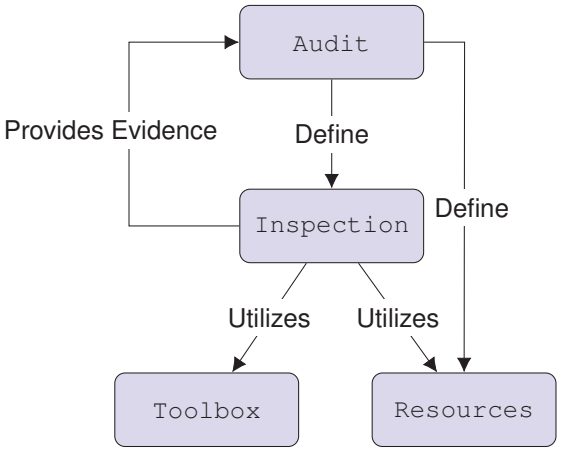

Fig. 5 Audit Framework Architecture Overview

the available resources and the necessary activities during the tests, so that the tools from the toolbox can be used to generate evidence.

C. ML Audit Workflow

The workflow shown in Fig. 6 begins with *specification*, in which the audit objectives, environment and constraints are defined, and the workflow is described as follows:

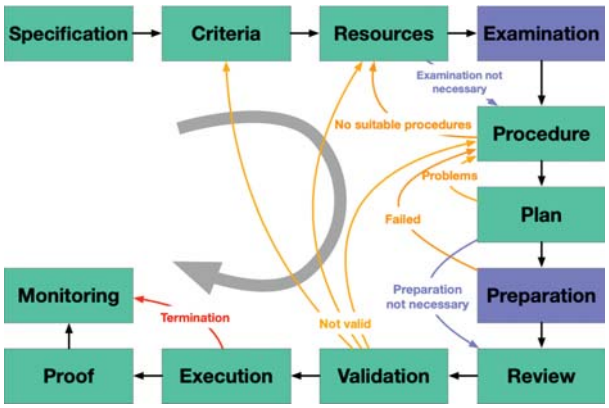

Fig. 6 Audit Workflow

1) *Criteria:* Defines the criteria of the data and model to be evaluated in the audit workflow. The criteria are evaluated to ensure adequacy, Inaccuracies are clarified.
2) *Resources:* Orchestrates the resources required to perform the audit tasks, taking into account the defined specifications. Examples of resources are the ML model and the data set.
3) *Examination:* If the specifications imply resource requirements, the audit task analyzes the specifications, criteria, and resources to ensure the validity of the audit results. This task is essential because faulty specifications, criteria, or resources would nullify and invalidate the audit results. The audit task also analyzes the structure of the model and resources in order to gain

insights that are necessary to determine the procedure defined in the next task. If the examination determines that the resources are insufficient to determine the audit procedure, the resource task is repeated, taking into account the results of the examination.

4) *Procedure:* Defines the procedure of the audit itself using the toolbox to provide the elements necessary to create the audit procedure. The structure of the procedure depends on the defined criteria.
5) *Planing:* Creates the order in which the audit is to be carried out, taking into account dependencies and resource coordination. A well defined plan ensures a smooth and efficient process. If no plan can be created due to conditions that cannot be met, the procedure must be changed.
6) *Preparation:* Consists of a preparatory action step for the final results and evidence. Examples would be the change of resources for the verification, the correction of defects. Depending on the test to be performed, preparation may not be necessary.
7) *Review:* At this point of the audit workflow all elements of the audit are defined. The results of the previous task are combined to create an executable audit.
8) *Validation:* The created executable audit is validated to ensure its validity, since errors in the audit would falsify the audit results. Defects discovered are corrected at their point of origin (criteria, resources or procedures).
9) *Execution:* The validated executable check is processed and the result is the result of the check. In case of non-compliance with a audit objective, there are several possible procedures: a) in case of a negative verdict for the model, the test could be aborted, b) jump directly to the report, or c) continue the audit as planned. In case of a positive verdict, the next task is continued.
10) *Proof:* The audit results are recorded and compared with the requirements, and detailed evidence is generated.
11) *Monitoring:* All generated evidence is collected, summarized and made available as a report. The granularity of the evidence report depends on the level of detail required.

D. Framework Automation and Implementation

The audit framework for ML models can automate many audit steps described in the audit workflow to minimize the audit effort and possible human error in the audit process. In this work, executable audits were created manually in Python, but evidence was generated, the audit and the audit report were processed automatically.

The framework uses the well known and proven machine learning frameworks keras [24] and tensorflow [25] to create its extensive functionality for the audit process.

The implementation of the audit framework (modules, structure and processes) was inspired by various existing audit frameworks. The framework developed by Holland et al. [26] to assign appropriate functions for providing data and generating results for each data set label has been adopted for use with the prediction model. The concept of augmenting datasets with coverage-guided fuzzing by Xie et al. [27] has been adopted for transforming a data set to generate an additional data set for test purposes. Burton et al. [28] argued that a predictive model fulfills a certain requirement if it provides a certain performance in an environment specified for a benchmark. The concept known as the security case pattern has been adapted for use in the audit framework. The audit framework was also developed by the architecture of Nishi et al. [3], which describes potential structures, tests and test types that can be tested within a machine learning product.

VI. Evaluation

For the evaluation of the functionality of the described audit ML framework the data set "Steel Plates Faults", which is part of the machine learning repository of the University of California, Irvine [29] was used. This data set contains 27 independent variables describing 7 types of steel plate failures. In our application example, we create a machine learning model to determine steel plate failure classes based on different steel plate properties. The audit objective of this use case is to determine the accuracy, robustness, and spurious relationship of our machine learning model and the occurrence of each defect class in the test data set. We define a set of rules for each metric to prove the conformity or violation of the model during the audit process.

Rule 1 Fault class proportion in test dataset

```
for each Class in fault classes do
    if Class proportion in dataset ≤ 10% then
        return False
    end if
end for
return True
```

Rule 1 defines that the test data set meets our requirements if each failure class has a share of more than 10% in the data set. Execution of the audit task shows that our machine learning model violates rule 1, as shown in Fig. 7.

To obtain meaningful and correct evidence in subsequent audit steps, the proportion of defect classes in the test data set is rebalanced by removing disproportionate test data from the data set.

Rule 2 Model accuracy

```
if Accuracy ≥ 72% then
    return True
else
    return False
end if
```

Rule 2 defines that the model meets our requirements if the accuracy is greater than or equal to 72%.

In the next step, the prediction of our model is compared to the prediction with the ground truth, and the results are quantified by the above mentioned metrics accuracy,

TABLE I
AUDIT RESULTS OF THE STEEL PLATES FAULT CLASSIFICATION

Inspection	Criteria	Metric	Value required	Value demonstrated	Assessment
Accuracy	C1	Accuracy	0.72	0.7201	Compliant
Accuracy	C1	F1-Score	0.7	0.732	Compliant
Robustness	R1	F1-Score	0.67	0.711	Compliant
Stability	R2	Neuron Coverage	0.85	0.85	Compliant
Stability	R3	Top-2 NC	0.7	0.625	Not compliant
Spurious relationship	S1	Plausibility rate	0.8	1	Compliant

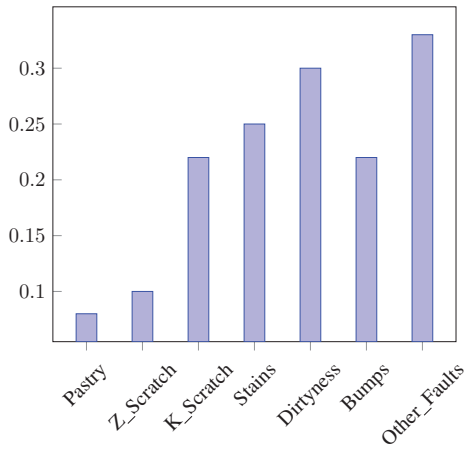

Fig. 7 Proportion of Fault Classes in the Test Data Set

robustness and spurious relationship. Auditing the robustness of the model requires creating an additional data set by transforming each record in the test dataset by replacing a random feature value with the average value for the feature over all records. Auditing the robustness also encompasses evaluate the relevance of the neurons within the model by determining the structural coverage metrics In the last step of the audit, the inference of the model is evaluated to determine whether a false relationship exists. The procedure consists of creating a local and linear approximation of the real inference.

The final step is the audit report, which consists of a summary of the audit task performed, as shown in Fig. 8, and the results for the audit itself for each criterion, as shown in Table I.

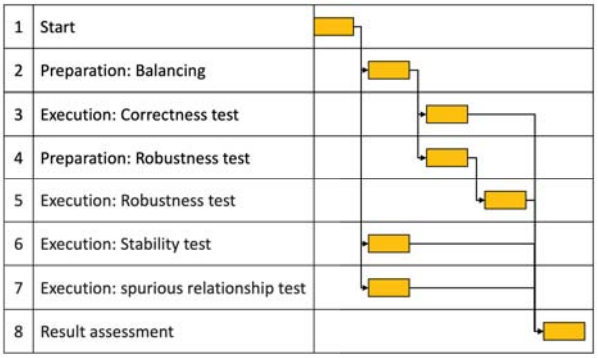

Fig. 8 Audit Report Showing the Audit Workflow

VII. CONCLUSION

An ML test frame with a corresponding workflow was designed and described. The framework facilitates the execution of audits for ML data, ML models and the ML development process. It defines the relevant elements, puts them into context and regulates the audit process. Due to the heterogeneity of ML models, the audit framework is abstract and not limited to specific model architectures or platforms. The applicability of this audit concept was evaluated on the basis of the use case: steel sheet defects.

REFERENCES

[1] "Künstliche intelligenz im mittelstand - relevanz, anwendungen, transfer," 2019, wissenschaftliches Institut für Infrastruktur und Kommunikationsdienste.
[2] W. Dai and D. Berleant, "Benchmarking contemporary deep learning hardware and frameworks: A survey of qualitative metrics," in *2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI)*, Dec 2019, pp. 148–155.
[3] Y. Nishi, S. Masuda, H. Ogawa, and K. Uetsuki, "A Test Architecture for Machine Learning Product," in *2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, Apr. 2018, pp. 273–278.
[4] U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J. M. F. Moura, and P. Eckersley, "Explainable Machine Learning in Deployment," p. 10, 2020.
[5] J. M. Zhang, M. Harman, L. Ma, and Y. Liu, "Machine Learning Testing: Survey, Landscapes and Horizons," *IEEE Transactions on Software Engineering*, pp. 1–1, 2020, conference Name: IEEE Transactions on Software Engineering.
[6] E. Stewart, K. Chellappan, S. Backhaus, D. Deka, M. Reno, S. Peisert, D. Arnold, C. Chen, A. Florita, and M. Buckner, "Integrated Multi Scale Data Analytics and Machine Learning for the Grid; Benchmarking Algorithms and Data Quality Analysis," Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), Tech. Rep., 2018.
[7] S. Schelter, D. Lange, P. Schmidt, M. Celikel, F. Biessmann, and A. Grafberger, "Automating large-scale data quality verification," *Proceedings of the VLDB Endowment*, vol. 11, no. 12, pp. 1781–1794, Aug. 2018. (Online). Available: http://dl.acm.org/citation.cfm?doid=3229863.3275547
[8] L. Barrett and M. W. Sherman, "Improving ML Training Data with Gold-Standard Quality Metrics," p. 4, 2019.
[9] G. S. Handelman, H. K. Kok, R. V. Chandra, A. H. Razavi, S. Huang, M. Brooks, M. J. Lee, and H. Asadi, "Peering Into the Black Box of Artificial I ntelligence: E valuation M etrics of Machine Learning Methods," *American Journal of Roentgenology*, vol. 212, no. 1, pp. 38–43, Jan. 2019. (Online). Available: https://www.ajronline.org/doi/10.2214/AJR.18.20224
[10] D. Rolnick, A. Veit, S. Belongie, and N. Shavit, "Deep learning is robust to massive label noise," 2018. (Online). Available: https://openreview.net/forum?id=B1p461b0W
[11] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," *CoRR*, vol. abs/1511.04599, 2015. (Online). Available: http://arxiv.org/abs/1511.04599

[12] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, May 2017, pp. 39–57.

[13] R. Salay and K. Czarnecki, "Using machine learning safely in automotive software: An assessment and adaption of software process requirements in ISO 26262," *CoRR*, vol. abs/1808.01614, 2018. [Online]. Available: http://arxiv.org/abs/1808.01614

[14] B. Waltl and R. Vogl, "Increasing transparency in algorithmic-decision-making with explainable ai," *Datenschutz Datensicherheit - DuD*, vol. 42, pp. 613 –617, Sep. 2018.

[15] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," 2019.

[16] "Global perspectivesand insightsartificial intelligence – considerations for the profession of int ernal auditing," The Institute of Internal Auditors. [Online]. Available: https://na.theiia.org/periodicals/Public Documents/GPI-Artificial-Intelligence.pdf

[17] C. Hutchison, M. Zizyte, P. E. Lanigan, D. Guttendorf, M. Wagner, C. Le Goues, and P. Koopman, "Robustness testing of autonomy software," in *2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*, 2018, pp. 276–285.

[18] K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore," *Proceedings of the 26th Symposium on Operating Systems Principles - SOSP '17*, 2017. [Online]. Available: http://dx.doi.org/10.1145/3132747.3132785

[19] M. Paschali, S. Conjeti, F. Navarro, and N. Navab, "Generalizability vs. robustness: Adversarial examples for medical imaging," 2018.

[20] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, I. Ray, N. Li, and C. Kruegel, Eds. ACM, 2015, pp. 1322–1333. [Online]. Available: https://doi.org/10.1145/2810103.2813677

[21] I. Myrtveit, E. Stensrud, and M. Shepperd, "Reliability and validity in comparative studies of software prediction models," *IEEE Transactions on Software Engineering*, vol. 31, no. 5, pp. 380–391, 2005.

[22] B. R. Aditya, R. Ferdiana, and P. I. Santosa, "Toward modern it audit- current issues and literature review," in *2018 4th International Conference on Science and Technology (ICST)*, 2018, pp. 1–6.

[23] F. Dölitzscher, T. Rübsamen, T. Karbe, M. Knahl, C. Reich, and N. Clarke, "Sun behind clouds - on automatic cloud security audits and a cloud audit policy language," vol. 06.2013, no. 1 & 2, pp. 1 – 16, 2013.

[24] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[25] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[26] S. Holland, A. Hosny, S. Newman, J. Joseph, and K. Chmielinski, "The Dataset Nutrition Label: A Framework To Drive Higher Data Quality Standards," *arXiv:1805.03677 (cs)*, May 2018, arXiv: 1805.03677. [Online]. Available: http://arxiv.org/abs/1805.03677

[27] X. Xie, L. Ma, F. Juefei-Xu, M. Xue, H. Chen, Y. Liu, J. Zhao, B. Li, J. Yin, and S. See, "DeepHunter: a coverage-guided fuzz testing framework for deep neural networks," in *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ser. ISSTA 2019. Beijing, China: Association for Computing Machinery, Jul. 2019, pp. 146–157. (Online). Available: https://doi.org/10.1145/3293882.3330579

[28] S. Burton, L. Gauerhof, B. B. Sethy, I. Habli, and R. Hawkins, "Confidence Arguments for Evidence of Performance in Machine Learning for Highly Automated Driving Functions," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2019, pp. 365–377.

[29] "Steel plates faults data set." (Online). Available: https://archive.ics.uci.edu/ml/datasets/Steel Plates Faults

Jan Stodt is a member of the Institute for Data Science, Cloud Computing and IT-security and a member of the faculty of computer science at the University of Applied Science in Furtwangen (HFU). He received his B. Sc. degree in computer science from the University of Applied Science in Furtwangen (HFU) in 2017 and his M. Sc. degree in computer science for the University of Applied Science in Furtwangen (HFU) in 2019.

Christoph Reich is professor (since 2002) at the faculty of computer science at the university of applied science in Furtwangen (HFU) and teaches in the field of network technologies, IT protocols, IT security, Cloud Computing, and Machine Learning. He is CIO of the HFU Information- and Media Centre, that is the scientific director for the IT data centre, Online-Services, Learning-Services, and library department. He is head of the Institute of Data Science, Cloud Computing and IT-Security (IDACUS; idacus.hs-furtwangen.de). Several founded research projects (FP7 EU, BMBF, MWK) have been accomplished successfully.