Towards Trustworthy AI Engineering - A Case Study on integrating an AI audit catalog into MLOps processes

Lennard Helmer
Claudio Martens
Dennis Wegener
Daniel Becker
{firstname.lastname}@iais.fraunhofer.de
Fraunhofer Institute for Intelligent Analysis and Information Systems
Sankt Augustin, Germany

Maram Akila
maram.akila@iais.fraunhofer.de
Lamarr Institute for Machine Learning and Artificial Intelligence
Sankt Augustin, Germany

Sermad Abbas
sermad.abbas@gmail.com

ABSTRACT

In recent years, Machine Learning Operations (MLOps) has become increasingly important as more and more Machine Learning (ML) based applications are brought into production. With this widespread, attention must be paid to the application's trustworthiness. Numerous methods and tools have already been developed in the area of trustworthy AI. However, the integration of those into the MLOps cycle and in particular into the pipeline engineering process is missing. To address this open problem, we analysed an AI audit catalog and translated the respective requirements into a healthcare IT service provider's MLOps process. In this work, we describe the translation process and present the insights obtained via a case study. Our work highlights the necessary considerations for professionals and the scientific community when dealing with similar challenges in Trustworthy AI engineering and operations and provides clear recommendations.

CCS CONCEPTS

• **Software and its engineering** → **Agile software development**.

KEYWORDS

MLOps, Machine Learning, Engineering, Trustworthy AI, Software Engineering, Development, Case study

ACM Reference Format:
Lennard Helmer, Claudio Martens, Dennis Wegener, Daniel Becker, Maram Akila, and Sermad Abbas. 2024. Towards Trustworthy AI Engineering - A Case Study on integrating an AI audit catalog into MLOps processes. In *2024 International Workshop on Responsible AI Engineering (RAIE '24), April 16, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3643691.3648584

RAIE '24, April 16, 2024, Lisbon, Portugal

ACM ISBN 979-8-4007-0572-4/24/04.
https://doi.org/10.1145/3643691.3648584

1 INTRODUCTION

Using machine learning in administrative applications in the healthcare sector can bring several benefits. It can automate manual administrative tasks, such as appointment scheduling and billing, leading to increased efficiency and reduced human error. Machine learning algorithms can analyse data to identify patterns and trends, helping healthcare organisations optimise their operations, resource allocation, and decision-making processes. Yet, the widespread use of machine learning models in software products, particularly in healthcare, has raised concerns about issues such as data privacy, security, and trustworthiness of these applications in general.

Lawmakers around the world are beginning to respond and introduce new laws governing the development and operation of AI, as evident by, e.g., the upcoming "EU AI Act"[1] or the US "Blueprint for an AI Bill of Rights."[2] Simultaneously, awareness of the potential damage of harmful AI applications is growing within society, putting further pressure on companies and development teams to consider trustworthiness when developing and operating machine learning applications.

In [Hussain et al. 2022] it is stated that human values such as privacy, transparency, integrity, social justice and diversity are becoming increasingly important in the development of software. It is also emphasized that design decisions inevitably have an impact on human values. Thus, it is important to consider these values as primary design considerations. The paper also mentions that companies are increasingly recognizing that considering such values is a key to business success. It points out that it is still unclear to what extent software development already takes these concerns into account and what challenges arise in doing so.

Thus, bringing together MLOps and Trustworthy AI in the healthcare domain has significant potential. MLOps, on the one hand, can help ensure the seamless development, deployment and operation of AI applications in healthcare settings, enabling continuous monitoring, updating, and performance optimization of ML models. Trustworthy AI, on the other hand, encompasses multiple dimensions such as transparency, fairness or accountability, which must be addressed to either avoid common problems such as unjustified discrimination or to achieve specific goals, e.g. explainability of AI decisions made to users or experts.

[1] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206
[2] https://www.whitehouse.gov/wp-content/uploads/2022/10/Blueprint-for-an-AI-Bill-of-Rights.pdf

While Trustworthy AI is already a vibrant area of research, for a survey see [Houben et al. 2022; Li et al. 2023], improvements often address specific and isolated problems, for instance adversarial robustness [Goodfellow et al. 2014] or membership inference [Shokri et al. 2017]. It is an ongoing challenge of how MLOps can adapt and integrate Trustworthy AI methodologies to its core to enable development teams to maintain the efficiency of their development processes while meeting the demands of legislators and society. For example, existing research on SecMLOps [Zhang and Jaskolka 2022], on ethical aspects of ML application [Lu et al. 2022] [Amugongo et al. 2023] and on trustworthiness [Li et al. 2023] structure Trustworthy AI methods along development life cycles but do not explicitly focus on preparing the resulting software for audits and are difficult to integrate into existing MLOps processes within an organisation.

In the work of [Muccini and Vaidhyanathan 2021], which discusses the current outlook of software architecture in the context of ML-based systems, it is noted that defining standard processes for the architecture of ML-based systems is essential given the ever-growing use of ML in production. This implies that extensive research is needed to understand how to better manage and document design decisions, taking into account, e.g., data quality, data volume, the probabilistic nature of learning algorithms, etc., as well as the limitations of software systems.

As part of a case study, we address these challenges and show a way in which the development of ML solutions in companies can be enhanced by establishing standard processes, such as the orientation along predefined decision records. In our approach, we work with practitioners to incorporate Trustworthy AI into their MLOps processes. Using an AI audit catalog as foundation, we identified, consolidated and summarised the relevant tasks. We analysed the catalog and mapped the requirements to an adapted MLOps process of the partner company. This allowed us to streamline the overall effort and offer the developers a clear solution. As result, we have extracted 82 consolidated tasks from the extensive catalog and applied them as part of the case study.

The remainder of this paper is organised as follows: Section 2 summarises background and related work on MLOps, Trustworthy AI, and AI auditing. In Section 3 we describe the MLOps structure as well as the goals and challenges of our case study on combining MLOps and Trustworthy AI engineering. Section 4 presents our approach on integrating AI audits into MLOps and in Section 5 we present and discuss the results of our case study. We conclude and give an outlook on future work in Section 6.

2 BACKGROUND AND RELATED WORK

In the following, we present background information on MLOps, Trustworthy AI, and AI auditing. Furthermore, we present related work.

2.1 MLOps

Artefacts that are created during the development of machine learning models, such as the model itself and its associated data processing pipelines, are highly dependent on the quality of the underlying data. Developing ML applications requires specialised skills and roles within the development team, as well as adapting a variety of new data exploration and versioning processes. On the one hand, software engineering paradigms like DevOps [Ebert et al. 2016], which enable software development teams to overcome common challenges of traditional development, lack processes to address the challenges introduced by the widespread adoption of machine learning projects. On the other hand, ML process models such as CRISP-DM [Wirth 2000], lack in covering the application scenario of ML models over a long period of time and in guidance on quality assurance [Studer et al. 2021]. To overcome the limitations of these paradigms, a new software development paradigm was introduced that specifically focuses on software including machine learning components: Machine Learning Operations (MLOps). Kreuzberger et al. [2023] offer a structured description of the related tasks, roles and components. The new paradigm builds up on well established processes, roles and tools known from DevOps and combines them with CRISP-DM. Similar to DevOps, the goal of MLOps is to reduce the development to production time, ensure high quality of artefacts, and a smooth cooperation between stakeholders. We refer to the MLOps cycle [Beck et al. 2020] as an iterative approach on MLOps that includes the following 6 phases: design, exploration, development, continuous integration, continuous deployment, and operations. It is shown in Figure 1.

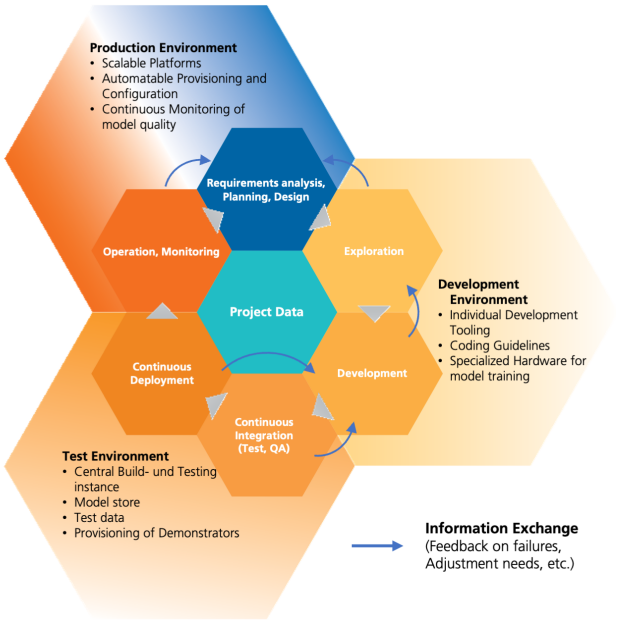

Figure 1: The MLOps cycle [Beck et al. 2020].

2.2 Trustworthy AI

With the increased use and performance of AI, the focus has shifted from "pure'' use to its trustworthiness, especially when the use or operation of the AI system is associated with potential loss, e.g. financial damage, discrimination, or harm of humans. Here, trustworthiness is an umbrella term encompassing multiple aspects or dimensions. The selection of those is driven both from the direction of ongoing research, with the discovery of new vulnerabilities or

shortcomings, as well as by upcoming regulations and emerging guidelines.

We follow the works of [Poretschkin et al. 2021], which structures trustworthiness into 6 broad categories:

(1) Fairness
Concerned with data bias and discrimination
(2) Autonomy and Control
Regards interplay between human and AI system
(3) Transparency
Explainability and tracability for users or experts
(4) Reliability
Performance, robustness, and generalisation issues
(5) Safety and Security
Interfaces with functional safety, integrity, and availability
(6) Data Protection
Privacy of personal data, data leakage in general

While there is, to date, no definite answer as to which dimensions belong to Trustworthy AI, variants of above aspects are found in most guidelines or regulations. However, separation into fields might differ. For instance, [Alzubaidi et al. 2023] sees robustness and accuracy, i.e. performance, as primary dimensions and not as sub-fields while including reproducibility and acceptance as new aspects. The HLEG recommendations,[3] which influenced the EU AI Act use a similar list as above, but with the addition of "societal and environmental wellbeing," i.e. by including aspects such as energy consumption or the influence of algorithmic recommendations in social networks. For an overview of already existing and upcoming regulations, see [Jobin et al. 2019], or more generally [Kaur et al. 2022; Thiebes et al. 2021].

2.3 AI Auditing

Auditing and certification are important tools for quality- and risk management of AI applications and to ensure the trust of stakeholders in ML applications and artefacts. Upcoming regulations like the EU AI Act demand specifically the assessment of conformity (Art. 49). To enable auditors to run a holistic assessment of the trustworthiness of an AI application, Fraunhofer IAIS published an AI audit catalog [Poretschkin et al. 2021]. It targets on one hand auditors to guide them through the testing process of existing AI applications and one the other hand developers to give them practical suggestions on how to build trustworthy AI applications.

AI audits usually entail numerous tests and quantitative assessments both of the model and the data (training, test, validation, and production). This can result in considerable expenses in terms of computational resources, might raise the practical problem of integrating lots of different test tools from various sources, and can be difficult to implement in an auditable fashion, especially when multiple independent parties are involved. To tackle these challenges, dedicated assessment platforms [Pintz et al. 2024] can enable reproducible, distributed, multi-party workflows and facilitate assessment automation by integrating them into MLOps pipelines.

[3]https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

2.4 Related Work

Research has already been conducted in the area of combining trustworthiness and MLOps. Zhang and Jaskolka [2022] introduce the SecMLOps paradigm, focusing on security as main driver for trustworthiness and enhance MLOps by incorporating different security measures into the processes and roles. Lu et al. [2022] conducted a literature review on ethical aspects of ML application development and structure their results in governance, process and system perspectives. Li et al. [2023] investigate the key aspects of trustworthiness and propose a systematic approach for development and evaluation. They also investigated the interaction between different trustworthiness aspects and how they influence each other. Amugongo et al. [2023] use a health care application as case study to describe possible measures to ensure trustworthiness and structure it along a AI development life cycle.

What is missing in existing research is the possibility to easily integrate trustworthy aspects into existing development processes like MLOps and the explicit focus on preparing ML applications for AI audits. We fill this gaps by integrating a well established AI audit catalog into an existing MLOps process.

3 CASE STUDY SETUP AND GOALS

In the following, we describe the setup of our case study, including the initial MLOps setup and the goals and challenges for integrating an AI audit catalog into a MLOps process.

3.1 Existing MLOps structure

Our project partner, a managed service provider for IT services in the healthcare sector, builds and operates a range of software products and data platforms for German health insurance companies. The company employs over 1500 people and generates annual sales of around 350 million euros. Software development projects within that company follow established guidelines that cover topics like budgeting, role definitions, and deployment procedures and usually adhere to the agile values and principles.

Although the level of experience in developing and operating traditional software products is high, a growing number of ML projects on automating customer processes and internal processes made it necessary to formalise a development process for ML software. This led to the introduction of MLOps as a development paradigm.

Standardisation of the development processes, documentation requirements, and tools aims at enhancing the quality of the developed software while reducing development time. Additionally, close collaboration between software developers, data scientists, and operators is anticipated to foster a growing level of experience within the company.

Based on the theoretical foundations of MLOps as described in Beck et al. [2020], the company introduced new project maturity levels for their ML projects that cover all organisational, technical and procedural aspects relevant to a project. These project maturity levels are visualized in Figure 2 and consists of the following levels: ideation, data evaluation, Proof-of-Concept (PoC), prototype, minimal viable product (MVP), pilot, and production.

Each project maturity level requires the fulfillment of certain MLOps tasks and quality measurements, e.g. for a project at the level prototype it is necessary to address the phases of the MLOps

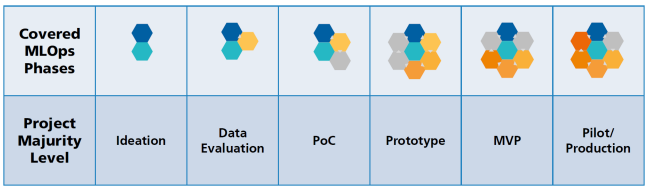

Figure 2: MLOps phases covered by project maturity levels, gray tiles represent optional phases

cycle up to the continuous integration (CI) phase in order to have automated CI processes up and running for the prototype.

The different project maturity levels provide orientation the project teams so that it is always clear which tasks they should focus on in the current level. Regular quality gates between levels enable the project team, management, and customers to terminate a project and minimise losses if the outcome fails to meet expectations. This prevents long-lasting projects that are unlikely to achieve a production-ready status and provides the project team with a clear road map and an overview of formal expectations. Every new machine learning use case must adhere to these project maturity levels to guarantee consistent high quality of the development process, the resulting artefacts and finally the product running in production.

In detail, each project maturity level covers a certain part of MLOps phases, depending on it's maturity.

The ideation level corresponds to the design phase in the MLOps model. It contains foremost theoretical considerations on the use case. The goal is to develop an idea for a solution and to gain acceptance of sponsors (management and/or customers) to continue the product development.

The data-evaluation level adds the exploration phase. Data from different health-insurance companies are often very heterogeneous and may come from variety sources. During the data evaluation a deep understanding of the data is gained and its quality is assessed for the use case. Especially health care data is of high criticality regarding the use of personal data. Therefore specific requirements both from a legal perspective (e.g. permissible use) as well as from the point of trustworthiness (e.g. protection of privacy) need to be met.

If the data quality is considered adequate for goal of the use case, the project goes into the PoC level. It covers the phases design, exploration and development. However, the development effort is mostly put into finding a model that can be presented to the stakeholders to decide on the business value.

A successful PoC starts the prototype level, where software-engineering practices play a major role for the development. The goal is to create a usable demonstrator and at least a concept on how the product can be integrated into existing business processes. This level adds tasks of the CI phase of the MLOps cycle by introducing automated pipelines to build and test the model.

When the prototype convinces with respect to usability and economic value, a MVP is developed. The main focus of the project team now lies on the software development and exploration becomes less important. In addition to dealing with the development and continuous integration, different deployment scenarios are evaluated to integrate a usable product into a testing environment. The MVP should contain rudimentary monitoring features. Especially for external customers the hardware infrastructure of the different insurance companies may vary, this level is used to carefully plan the deployment processes for production with each customer individually.

For a final quality control, e.g. in terms of performance and fault tolerance, the product is deployed to pilot environments. This is particularly important as many ML products will be used inside complex existing systems. A successful pilot level leads to the production level where the product is used in the daily business routines. Monitoring is now an integral part of the product to ensure the ongoing reliability of the product and to spot potential problems such as, e.g. drifts or errors.

While the company's project maturity levels do not involve going back to a previous level, it may be necessary to switch back and forth between the different MLOps phases that are covered by this level. For example, when developing the prototype, it might be necessary for the project team to gather a deeper understanding of underlying business or put more effort in exploring the data to construct a better model.

3.2 Goals and challenges

With a growing amount of machine learning projects on the roadmap and upcoming regulations like the EU AI Act, the company decided to enhance their project maturity levels and define measures that are meant to ensure the trustworthiness of their ML applications.

Together, we defined the following goals for the adapted project maturity levels:

(1) Guideline for project teams
(2) Minimal additional effort
(3) Prepare products for trustworthiness audits
(4) Strengthen stakeholder trust in products with ML functionalities

Encouraging the right decisions at the right time by providing guidance for teams can greatly improve the efficiency of plannings and structure the development plan. For each of the different project maturity levels the project teams should have a guideline that supports them in prioritising Trustworthy AI decisions when they become important and in being able to plan related tasks in future development sprints. Furthermore it can support an informed and structured communication with product owners and other decision makers. Our solution is meant to support project teams and integrate into existing processes that support efficient development. Thus, any solution must minimise the additional effort as much as possible. Prioritising the ability to audit an ML applications while the development is still ongoing ensures that no or only minimal future adaptations are necessary when audit requirements are coming up. Apart from audit requirements, trust and data protection are of high value as the stakeholders work with sensible healthcare data in a strongly regulated field and value transparent data-protection measures. While the benefits of using ML for enhancing healthcare services are obvious, each project has to prove that it is not only achieving monetary and performance goals, but also respects aspects of Trustworthy AI.

A well-structured and transparent process that documents which Trustworthy AI measures were taken, the reasoning behind it, and the effects that it had on the overall system strengthens the trust of stakeholders, like healthcare insurance companies, and their policyholders.

4 INTEGRATING AI AUDIT INTO MLOPS

To achieve the goals we adapted the existing structures and processes, if required, on condition that the adaptation supports the development teams with clear action steps but keep the necessary additional work and complexity to a minimum.

After reviewing the current state of research, we decided to do this on the basis of the AI audit catalog by Poretschkin et al. [2021]. The catalog presents the checks that are carried out during an audit for trustworthiness and therefor provides insights into the requirements an application has to fulfill to be considered trustworthy. However, this catalog is intended for an audit situation after completion of the primary development process. As we want to support the development process itself, we have decided to translate the requirements from the AI audit catalog into tasks for the development.

Together with experienced employees from our partner, who have in-depth knowledge of all current ML use cases within the company, we analyzed each Trustworthy AI dimension as described in the AI audit catalog to identify potentially relevant tasks for each of the project maturity levels. The experience of the company representatives ensured that the company's current approach could be efficiently supplemented with the actions from the Trustworthy AI area. Those measures were translated into tasks and, considering the large amount of topics covered by the AI audit catalog, summarized if possible to ensure minimal additional effort and avoid excessive demands for the product teams. Those tasks were then assigned to one of the project maturity levels based on internal prioritization, processes, and requirements.

The company follows agile development practices and project teams work with a high degree of self responsibility. The requirement of the existing project maturity levels is to consider certain topics when the dedicated level is reached, decide upon an appropriate approach and document the results. The same fundamental idea was used for new tasks which were identified during our analysis of the AI audit catalog.

An illustration of the process used to obtain the results, which is explained in more detail below, can be seen in Figure 3.

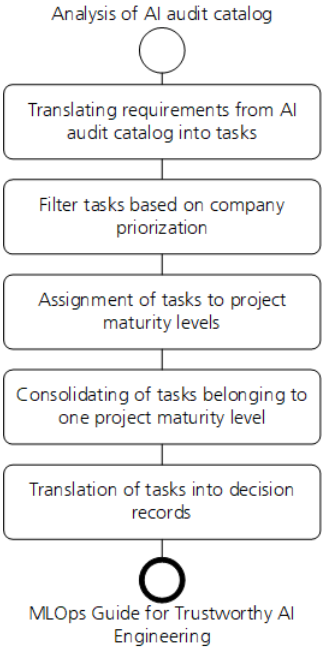

Figure 3: Process diagram

After reviewing the AI audit catalog, it became evident that it contained a substantial number of requirements, in total 242. Compiling and presenting them to the developers as necessary tasks would be impractical. To provide developers with a clearer solution, we decided to summarize the requirements whenever possible and exclude those deemed irrelevant to the company's processes. This approach prevents developers from being overwhelmed by confusion or excessive effort if they were to use our results as a guide in the future.

The first step was to convert the requirements from the AI audit catalog into tasks. It was important to ensure that the derived tasks were not audit tasks, but tasks that could be carried out during development process. In addition, we examined the individual requirements within each of the six dimensions of the AI audit catalog, which encompass risk analysis, criteria, and measures, to determine their relevance to the companies use cases. Upon evaluation, we found that none of the points were deemed irrelevant, and therefore, all requirements were retained. In order to present the results more clearly, we connected the tasks to the project maturity levels known to the company. The next step was to consolidate tasks that belong to the same project maturity level based on similar task content. We recognized that the derived tasks, e.g. "Analyse your training data for bias", could be translated into decisions (e.g "Could bias be an issue considering your training data? And if yes what steps are necessary to mitigate the risk?") that the team will face during the development. This relates to goal 3 to ensure the ability to audit the resulting software for trustworthiness which demands that a well defined documentation process is implemented.

To address both topics we decided to transfer the tasks into decision records. Decision records are an established tool to document decisions in a structured way and store them for later reference. We followed a template suggested by Van Heesch et al. [2012] and adapted it for our needs as seen in Table 1. For each decision the id, decision status, date, decision group, context/issue, decision, arguments, method, related requirements, related artefacts and project maturity levels have to be documented.

Finally, we transferred the tasks into our MLOps guide for trustworthy AI Engineering. The specific results that emerged are discussed in detail in the next chapter.

Table 1: The structure and fields of one decision record

Decision record structure	
ID	Unique identifier for decision record
Status	Decision status, either accepted/open/rejected
Decision group	Persons who were involved in the decision making
Context/ Issue	Background information, questions, description of the problem
Decision	Record of the decision
Arguments	Description of the pro and contra arguments
Method	Documentation of the methods used, analysis tools, etc.
Related Requirements	Related requirements and decision records
Related Artefacts	Related artefacts or additional documentation that was created during the process
Project Maturity Level	The project maturity level to which the decision belongs or was taken in

5 CASE STUDY RESULTS

In the following, we present the main contributions of our work and discuss the results.

We have extracted a list of 82 consolidated tasks from the 242 requirements in the AI audit catalog. Those requirements were structured, assigned to levels of the project maturity levels, and translated to decision records. The decision records will guide the development of future software by prioritizing important decisions at the right level of the project maturity levels.

By translating the AI audit requirements in decision records we not only support the planning and forecast of upcoming tasks but also ensure the traceability of decisions, related to Trustworthy AI, in retrospective by enforcing a structured documentation of the decision itself and the argumentation that led to this decision. This positively influences the auditability of the resulting AI application as development teams can answer questions that they would face during an audit related to fairness, security, autonomy and control, data protection, explainability and robustness.

We evaluated our approach by implementing the results into two recently started AI development projects at our partner company. The results are promising and emphasise the usability. During the design sprint of one of the projects, the team, guided by the provided decision record templates, identified that the planned AI driven software reaches level three of four possible levels in regard of the AI autonomy and that the level of human control is limited. This realisation led to a discussion about possible dangers and strategies to mitigate the risk. The decisions that were made during this early level influenced the proposed architecture and will be considered in the future development.

By considering the aspects of trustworthy AI, the teams had the possibility to think about the use case, solutions, and the consequences of decisions from a new perspective.

The guide will be continuously evaluated for later project maturity levels, which then cover more phases of the MLOps cycle. Even if these later levels are more complex, we hold the belief that the guide can provide similar assistance and support as it did in the initial level.

Our approach is expected to demonstrate its capability in supporting a comprehensive audit of a production-level AI application. The validation of this will be conducted in the future when use cases progress through the different project maturity levels and first audits can be conducted.

Furthermore, it is important to encourage active participation from additional project teams to gather valuable feedback.

Our work on integrating an AI audit catalog into existing MLOps processes is valuable not only for our project partner but also for other organisations facing similar challenges. It can be easily adapted and integrated into other MLOps processes. The lightweight approach of using decision records preserves the self responsibility of development teams and motivates the planning and integration of Trustworthy AI topics at the right time.

6 CONCLUSION AND OUTLOOK

With the rising use of machine learning models in real-world applications, trustworthiness becomes increasingly vital. This is due to the significant impact these models have on decision-making processes, the need to address ethical concerns and biases, the importance of transparency and explainability, the requirement for data privacy and security, and the establishment of accountability and governance mechanisms.

So far, trustworthiness aspects have not yet been specifically integrated into the life cycle management of ML applications (MLOps). In this work, we presented a method for integrating the requirements of an AI audit catalog into a company's MLOps processes. As part of a case study, we showed how to integrate these requirements based on decision records into a company's MLOps processes. By translating the AI audit requirements into these decision records, we support the planning and forecast of upcoming tasks and also ensure the traceability of decisions for audit purposes. Finally we offer a roadmap on how to approach this topic for organisations in a similar situation.

With the rapidly growing relevance of Trustworthy AI, further research is needed on theoretical and practical challenges with regards to the integration of trustworthiness into MLOps processes. This includes, e.g. the development of concrete components that can be included into ML application to address trustworthiness requirements - up to automated monitoring processes to constantly ensure meeting audit requirements in production.

ACKNOWLEDGEMENTS

This research has been funded by the Fraunhofer Cluster of Excellence Cognitive Internet Technologies.

REFERENCES

Laith Alzubaidi, Aiman Al-Sabaawi, Jinshuai Bai, Ammar Dukhan, Ahmed H Alkenani, Ahmed Al-Asadi, Haider A Alwzwazy, Mohamed Manoufali, Mohammed A Fadhel, AS Albahri, et al. 2023. Towards Risk-Free Trustworthy Artificial Intelligence: Significance and Requirements. *International Journal of Intelligent Systems* 2023 (2023), 1–41.

Lameck Mbangula Amugongo, Alexander Kriebitz, Auxane Boch, and Christoph Lütge. 2023. Operationalising AI ethics through the agile software development lifecycle: a case study of AI-enabled mobile health applications. *AI and Ethics* (2023), 1–18.

Niklas Beck, Claudio Martens, Karl-Heinz Sylla, Dennis Wegener, and Alexander Zimmermann. 2020. Zukunftssichere Lösungen für maschinelles Lernen. (2020).

Christof Ebert, Gorka Gallardo, Josune Hernantes, and Nicolas Serrano. 2016. DevOps. *IEEE Software* 33, 3 (May 2016), 94–100. https://doi.org/10.1109/ms.2016.68

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

Sebastian Houben, Stephanie Abrecht, Maram Akila, Andreas Bär, Felix Brockherde, Patrick Feifel, Tim Fingscheidt, Sujan Sai Gannamaneni, Seyed Eghbal Ghobadi, Ahmed Hammam, et al. 2022. Inspect, understand, overcome: A survey of practical methods for ai safety. In *Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety*. Springer International Publishing Cham, 3–78.

Waqar Hussain, Harsha Perera, Jon Whittle, Arif Nurwidyantoro, Rashina Hoda, Rifat Ara Shams, and Gillian Oliver. 2022. Human Values in Software Engineering: Contrasting Case Studies of Practice. *IEEE Transactions on Software Engineering* 48, 5 (2022), 1818–1833. https://doi.org/10.1109/TSE.2020.3038802

Anna Jobin, Marcello Ienca, and Effy Vayena. 2019. The global landscape of AI ethics guidelines. *Nature Machine Intelligence* 1, 9 (2019), 389–399.

Davinder Kaur, Suleyman Uslu, Kaley J. Rittichier, and Arjan Durresi. 2022. Trustworthy Artificial Intelligence: A Review. 55, 2, Article 39 (jan 2022), 38 pages. https://doi.org/10.1145/3491209

Dominik Kreuzberger, Niklas Kühl, and Sebastian Hirschl. 2023. Machine learning operations (mlops): Overview, definition, and architecture. *IEEE Access* (2023).

Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. 2023. Trustworthy AI: From Principles to Practices. 55, 9 (2023), 1–46. https://doi.org/10.1145/3555803

Qinghua Lu, Liming Zhu, Xiwei Xu, Jon Whittle, and Zhenchang Xing. 2022. Towards a roadmap on software engineering for responsible AI. In *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*. 101–112.

Henry Muccini and Karthik Vaidhyanathan. 2021. Software architecture for ML-based systems: What exists and what lies ahead. In *2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN)*. IEEE, 121–128.

Maximilian Pintz, Daniel Becker, Michael Mock, and Maximilian Poretschkin. 2024. PARMA: a Platform Architecture to enable Automated, Reproducible, and Multi-party Assessments of AI Trustworthiness. In *2024 International Workshop on Responsible AI Engineering (RAIE '24), April 16, 2024, Lisbon, Portugal.*

Maximilian Poretschkin, Anna Schmitz, Maram Akila, Linara Adilova, Daniel Becker, Armin B Cremers, Dirk Hecker, Sebastian Houben, Michael Mock, Julia Rosenzweig, et al. 2021. Leitfaden zur Gestaltung vertrauenswürdiger Künstlicher Intelligenz (KI-Prüfkatalog). (2021).

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 3–18.

Stefan Studer, Binh Bui, Christian Drescher, Alexander Hanuschkin, Ludwig Winkler, Steven Peters, and Klaus-Robert Müller. 2021. Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. *Machine Learning and Knowledge Extraction* 3 (04 2021), 392–413. https://doi.org/10.3390/make3020020

Scott Thiebes, Sebastian Lins, and Ali Sunyaev. 2021. Trustworthy artificial intelligence. *Electronic Markets* 31, 2 (2021), 447–464.

Uwe Van Heesch, Paris Avgeriou, and Rich Hilliard. 2012. A documentation framework for architecture decisions. *Journal of Systems and Software* 85, 4 (2012), 795–820.

R. Wirth. 2000. CRISP-DM: Towards a Standard Process Model for Data Mining. In *Practical application of knowledge discovery and data mining, Proceedings of the International Conference on the Practical Application of Knowledge Discovery and Data Mining*. Practical Application Co;, 29–40. https://www.tib.eu/de/suchen/id/BLCP%3ACN039162600

Xinrui Zhang and Jason Jaskolka. 2022. Conceptualizing the Secure Machine Learning Operations (SecMLOps) Paradigm. In *2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS)* (2022-12). 127–138. https://doi.org/10.1109/QRS57517.2022.00023