Towards a structured AI development lifecycle for reusable AI products in the public sector

Albana Celepija[1,3], Bruno Lepri[2] and Raman Kazhamiakin[1]

[1]*Digital Society Center, Fondazione Bruno Kessler, Trento, Italy*
[2]*Center for Augmented Intelligence, Fondazione Bruno Kessler, Trento, Italy*
[3]*University of Trento, Trento, Italy. albana.celepija@unitn.it*

Abstract

Currently, AI-enabled solutions in the public sector are integrated in a fragmented way that does not allow them to be easily transferable to other public organisations, highlighting the need for methodologies that streamline AI development process. For this reason, regulatory bodies have established standards that define what characteristics AI systems need to meet. Meanwhile, recent research initiatives call for comprehensive methods that offer guidance on how to embed requirement aspects into the main AI development stages. However, existing approaches offer high-level recommendations overlooking the difficulties in translating them into low-level implementations in order to bring together procedural and technical approaches. Moreover, the level of granularity of AI lifecycle stages does not facilitate the modularity of AI systems, which would enable the easy integration of new implementation to meet system requirements. In this paper we enhance the AI lifecycle pipeline with explicit operations that aid in the modular development and reuse of AI systems. Furthermore, we propose a structured framework that enables the operationalisation of system-wide objectives throughout the various phases of the AI lifecycle, while systematically leveraging existing toolkits. To illustrate the framework's adoption, we present a proof-of-concept example to showcase its practical application.

Keywords

Reusable AI product, Operationalise AI, Enhanced MLOps, Public administration,

1. Introduction

Artificial Intelligence (AI) is experiencing rapid growth and demonstrates significant impact across various sectors, including public administration. AI capabilities improve decision-making processes, enhance operational efficiency, and foster public service delivery [1]. As AI continues to be adopted in public sector, its importance in driving innovation and creating economic value becomes evident. A distinguishing characteristic of applying AI in public administration domain is the similarity of the processes and operations across different organizations, which facilitates the reuse of AI-based solutions and stimulates their adoption.

Reaching this objective, however, remains a challenging problem. On the one side it is necessary to deal with the intrinsic complexity of delivering AI-based solutions reated to continuosuly evolving data, the unpredictable nature of AI models leading to variable outcomes, technological complexity. On the other side public administration sector to a much greater extent has face such problems as the lack of expertize and skill gaps, organizational implementation challenges, and insufficient architectural support on designing, deploying and monitoring AI-based systems.

Moreover, the development and deployment of AI products in public administration requires considering regulatory compliance and AI trustworthiness aspects[2]. Public organizations often operate without well-defined procedures to systematically develop, validate and automate the evaluation of critical characteristics such as privacy, fairness, resource constraints, and performance quality in line with the existing regulations and policies. Making these actions explicit, traceable and accountable, associating them to the regulatory requirements is fundamental for public administration in order to bring the AI-based innovation to its processes.

Current state-of-the-art approaches address these challenges from two perspectives. First, international AI regulatory authorities focus on establishing guidelines [3] and principles [4] to address

MLOps25: Workshop on Machine Learning Operations | co-located with ECAI 2025, Bologna, Italy

specific aspects of AI development and deployment, such as transparency, adaptability, and ethical considerations. This includes the definition of Responsible AI (RAI) practices [1] and the use of AI evaluation methodologies like Z-inspection [5] to ensure AI systems are transparent, accountable, and aligned with ethical standards. While these guidelines define the well-structured canvas for structuring and implementing compliant AI-based solutions, they remain at high level of abstraction making it difficult to operationalize them without necessary expertize.

Second, different solutions to support the complementary activities for AI product development and execution. These solutions, in the form of software libraries and framework may help in different areas, dealing with data and model quality, security and fairness, addressing different AI domains and applications. Still, their usage is often fragmented or too specialized; their relation to the regulatory settings and requirements is not often clear. Such clarity would help AI practitioners, and also non-AI experts, to seamlessly incorporate these tools into the lifecycle of an AI product in a modular and reusable manner, enabling both new and existing systems to be easily assembled according to initial or additional requirements.

To address these issues, MLOps practices should be enhanced to align more explicitly with system-wide requirements. Although extensive tools and frameworks have emerged on implementing individual ML pipeline components, developing production systems that incorporate them requires a more holistic approach which is still missing in the literature. It is essential to carefully plan the necessary AI lifecycle components and determine how they interconnect to support the system's overall functional and performance goals.

In this paper, we propose a revision of the reference ML operations (MLOps) life-cycle [6]. First, we enrich the life-cycle model with explicit and well-defined operations that have clear execution semantics and facilitate the automation of the underlying process. Second, we demonstrate how relevant cross-cutting requirements and aspects, such as quality, fairness, optimization, adaptability, privacy and transparency may be operationalized and aligned with the enhanced AI lifecycle operations. Third, we show how the existing implementation libraries and tools may be associated with this taxonomy, paving the way for modularity and reuse of these solutions across different contexts. Finally, we showcase how they can be implemented using existing solutions and tools facilitating the creation of reusable operations, modules, and even pipelines in different domains.

The paper is structured as follows. Section 2 provides some background information about the concepts that our paper relies on. In Section 3 we outline the framework components and its final representation. Section 4 describes a proof-of-concept on how to use the framework, and finally conclusions are drawn in Section 5.

2. Related work

Reusability of AI solutions in the public sector is an important aspect as highlighted in recent guidelines [2]. Various countries have adopted the practice of sharing software within the public sector to transform their administrative procedures. Initiatives like the E-Government Action Plan 2016-20202 [3], and Sharing and Reuse of IT solutions Framework [4] encourage member nations to reuse information and solutions. This approach aims to reduce costs, and improve the development of digital services by reusing existing components or entire solutions in a more transparent and organised way [7]. These guidelines and initiatives emphasize the importance of structured approaches to AI development, similar to traditional software development processes, but adapted to the unique challenges of ML and AI systems [8].

[1] https://msblogs.thesourcemediaassets.com/sites/5/2022/06/Microsoft-Responsible-AI-Standard-v2-General-Requirements-3.pdf

[2] https://www.agid.gov.it/sites/agid/files/2024-05/lg-acquisizione-e-riuso-software-per-pa-docs_pubblicata.pdf, https://www.agid.gov.it/sites/agid/files/2024-07/Strategia_italiana_per_l_Intelligenza_artificiale_2024-2026.pdf

[3] https://digital-strategy.ec.europa.eu/en/policies/egovernment-action-plan

[4] https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/2017-10/sharing_and_reuse_of_it_solutions_framework_final.pdf

Therefore, addressing the bespoke development of AI solutions in the public sector requires responsible development practices [9] to fully leverage AI's capabilities while minimizing risks. The effective implementation of AI relies on MLOps paradigm, which extends software engineering and DevOps principles to automate workflows and manage the AI lifecycle systematically. However, such continuous practices are not immediately compatible with regulatory requirements that often need authority involvement.

To this end, on one hand, a range of procedural tools to regulate AI [10, 3] have emerged, especially in public sector, due to stringent regulations. They offer high-level recommendations on the characteristics that an AI system must satisfy. On the other hand, from a practical perspective, AI industry and private sector have developed a series of technical tools to implement these regulatory aspects, focusing on individual AI lifecycle stages separately [11]. The following list includes some of these tools.

- Quality: frictionless [5]
- Fairness: fairlearn [6], AIFairness360 [7], holistic.ai [8]
- Optimisation: bitsandbytes [9]
- Adaptability: evidently.ai [10], alibi-detect [11], transformers [12]
- Transparency: model-card-toolkit [13]
- Privacy mostly.ai [14], giskard [15]

Although advancements are being made in creating regulatory, procedural, and technological tools, integrating trustworthiness, ethical considerations, and service-level requirements into the AI development process remains a challenge. This is being addressed by developing frameworks that assist in converting high-level requirements into practical steps. Frameworks like Z-inspection [5] and capAI [12], for instance, outline a multi-stage evaluation process throughout various phases of the AI lifecycle, encouraging the participation of diverse stakeholders and generating system-specific conformity assessments and recommendations.

Research efforts to integrate trustworthiness into the development of AI systems often treat each aspect of trustworthiness separately. Although the TOP methodology [13] demonstrates advantages by using documentation cards alongside risk management to evaluate AI system trustworthiness iteratively, it fails to offer a holistic perspective on the combined characteristics of trustworthiness. Other methods, like capAI [12] and POLARIS [14], try to bring together procedural and technical safeguards but overlook the level of granularity of the AI lifecycle due to the absence of standardized AI development processes. Meanwhile, approaches such as ECCOLA [15], TAII [16], and OOD-BC [17] fail to connect with existing toolkits in current state of the practice. These factors reduce the likelihood of implementing AI solutions that comply with regulations [18]. Table 1 summarizes a comparison between state of the practice frameworks that help translate AI requirements and principles into practice.

In sum, there is no existing framework or approach that offers a strategy to align actionable AI lifecycle operations with all the corresponding requirements an AI system must adhere to [19]. Moreover, there is a low level of automation of requirements implementation and verification due to a lack of standardization from a lifecycle perspective. There is still no structured and modular approach to leveraging existing technical tools which address these requirements in a fragmented way.

This paper proposes a framework, dedicated to implement a structured approach for requirements implementation, ensuring ethical functioning and adherence to service level agreements. Furthermore,

[5] https://framework.frictionlessdata.io/

[6] https://fairlearn.org/main/auto_examples/plot_credit_loan_decisions.html#sphx-glr-auto-examples-plot-credit-loan-decisions-py

[7] https://github.com/Trusted-AI/AIF360

[8] https://holisticai.readthedocs.io/en/latest/getting_started/index.html

[9] https://huggingface.co/docs/bitsandbytes/main/en/index

[10] https://www.evidentlyai.com/

[11] https://docs.seldon.io/projects/alibi-detect/en/stable/

[12] https://huggingface.co/docs/transformers/en/index

[13] https://www.tensorflow.org/responsible_ai/model_card_toolkit/guide

[14] https://github.com/mostly-ai/mostlyai

[15] https://docs.giskard.ai/en/latest/getting_started/index.html

Table 1
Methodologies to operationalize the requirements during AI system development process. X means that the dimension is not foreseen.

Methodology name	Alignment with AI lifecycle	Granularity of lifecycle	Outcome	Alignment with requirements	Implementation modularity level
TOP[13]	Yes	High-level: stage	Cards, Risk management, Toolkits	Partial	Low
capAI [12]	Yes	High-level: stage	Socio-tech Recommendations	Holistic	Low
Z-inspection [5]	No	X	Ethics Evaluation Report	Holistic	X
Eccola [15]	No	X	Ethical thinking cards	Partial	X
POLARIS [14]	Yes	Detailed	Knowledge sources guidelines	Holistic	Low
OOD-BC [17]	Yes	High-level: stage	Trustworthiness assessment	Holistic	X
TAII [16]	Yes	High-level: stage	Management guidance	Holistic	X
Our framework	Yes	Detailed: operations	Cards, Technical implementation	Holistic	High

it combines enhanced AI development operations with system requirements into a unified framework to systematically integrate existing technical tools that address some aspects in isolation toward achieving system-wide objectives.

3. The structured framework for the taxonomy of the AI development lifecycle

The proposed framework aims at modeling the operations involved in the development, deployment, monitoring and maintenance of an AI product in a structured way. Specifically, we represent the AI life-cycle phases, the horizontal axis, as a minimal set of explicit operations having their specific semantics and characteristics. Note that in a specific AI solution, each operation may have multiple implementation variants or even be omitted within the corresponding phase. We then refine these operations across various cross-cutting aspects, vertical axis, related to the established requirements that the AI system must satisfy and consider, such as quality, fairness, privacy, and others. In this way, we make the implementation of these aspects explicit in the life-cycle and provide a way to map them to concrete solutions and tools.

In this section, we explain the significance and role of both axes along with their respective components, in Section 3.1 and Section 3.2. Finally, in Section 3.3 we present the layout of the proposed framework which hosts the procedures that the AI solution implements.

3.1. Enhanced AI lifecycle operations

Building on earlier research on AI development lifecycle [20, 6, 21], we utilize the pre-established pipeline comprising the three high-level stages such as, data preparation, modeling, and operationalisation. Each stage in the lifecycle of an AI system consists of a set of operations that are common across different types of AI products [22, 23, 24]. Our goal is to establish a standardized set of operations with clear semantics, inputs and outputs that enable automation of the AI lifecycle, which in turn shortens the time to deployment and fosters AI adoption rates in the public sector.

Table 2
List of artifacts consumed and produced by AI lifecycle operations

Type	Description	Format	Examples
Data	The primary artifact fed into the training algorithm to fit the best model	Parquet, CSV, JSON	Training data, Production data, Evaluation data
Report	Structured information about the results obtained after applying a function	JSON, CSV	Data profiling reports, Metrics reports
Model	Either a single file or multiple files constituting the model	Pickle, pytorch, keras	Model artifacts
Configuration	Declarative specifications used to orchestrate the execution of individual components or pipelines	JSON, YAML	Hyperparameters, Criteria for evaluation, Metadata, Thresholds, Test cases
Status	A blocking artifact that defines the pipeline execution	Boolean, Alerts	True/False results when validating data or model before deployment
Documentation	Files that contain human-readable content	Text, Markup	Model Card [25], Data Card [26], Risk Card [27], AI Product Card [28], AI Cards [29]
Function	An implementation function	Python, C++	Data validation function, model training function, model serving function
Service	Service encapsulating the model and parameters for serving it, matching evaluation parameters	REST, App	Model serving endpoint or API
Logs	Generated during each phase of the AI lifecycle. Logging of events or Observability services	Prometheus	System-level monitoring: payload size, processing time, resource allocation

In defining the operations semantics, we first specified the list of artifacts that each operation expects as input and produces as output, which are summarized in Table 2. In AI systems, managing these artifacts, produced by various procedures, enables automation, supports quality assurance, improves reproducibility, and facilitates auditability.

Furthermore, to facilitate the development and reuse of an AI solution or its components, it is essential to ensure that the system is sufficiently modular. Modularity contributes also to easily 'apply' system requirements and in assessing whether these requirements are met. We achieve modularity by dividing the three main stages of the AI development lifecycle into 15 explicit operations, as shown in Figure 1. These operations serve as the fundamental unit of development and deployment of an AI system and may be revisited and executed iteratively in varying orders [30].

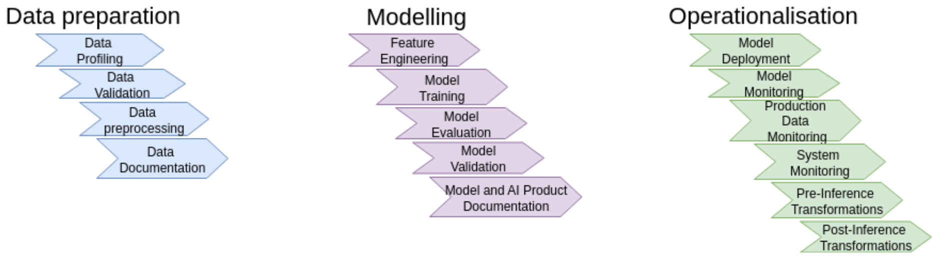

Figure 1: The stages of AI lifecycle

The first stage of the AI lifecycle is composed of four well-defined operations related to data preparation such as, data profiling, data validation, data preprocessing and data documentation. Data preprocessing in particular, may involve many code implementations that apply trasformations to data, including data cleaning, bias mitigation, or data augmentation. During data augmentation, the initial dataset undergoes modifications to increase its volume and diversity. It is commonly used when training data quantity is not sufficient, when some features contain bias or are not fairly distributed among different subgorups, or to address privacy concerns by avoiding the use of real data that contains sensitive information.

Then, the modeling stage includes the following operations: feature engineering, model training, model evaluation, model validation, and model and AI product documentation. Note that each operation may have multiple implementations within an AI solution, or different procedures addressing various system requirements may be grouped under the same operation category. Despite differing implementations, all variants within a given operation category share the same semantics, they receive inputs of the same kind and produce consistent type of outputs. This uniformity supports the modular design principle inherent in our proposed framework. Also, this means that each implementation of an operation in the AI pipeline functions as a small piece of the broader AI system, collectively contributing to the system's overarching objectives.

Finally, the operationalisation stage consists of operations such as, model deployment, model monitoring, production data monitoring, system monitoring, pre-inference monitoring, and post-inference monitoring. Specifically, model deployment involves getting the model ready for the serving phase, enabling it to be accessed and utilized for inference, generation, classification, or the particular task it was designed for. It is crucial to note that the result of deployment is a function that encapsulates the model along with its parameters and settings required for its operation. These parameters are identical to those used during the model evaluation and validation stages, ensuring that the deployed model is the same one that was previously assessed. Table 3 summarizes the minimal list of explicit operations that make up the stages of the AI development lifecycle together with their input and output artifacts we defined previously.

Table 3: The list of operations that make up the stages of the AI development lifecycle

Stage	Operation	Description	Input	Output
Data Preparation	Data Profiling	Generate reports with specific characteristics of training data	-Data	-Report
	Data Validation	Validate the quality of training data to identify issues that could impact model performance	-Data -Configuration	-Status
	Data Preprocessing	Apply data cleaning procedure, data augmentation, type conversion, analyze data distribution, evaluate bias or discrimination issues on data	-Data	-Data
	Data Documentation	Generate human readable documentation in order to facilitate knowledge transfer and increase transparency	-Data	-Documentation
Modelling	Feature Engineering	Apply principal components analysis, understand feature importance	-Data	-Data -Report
	Model Training	Training procedure that fits the model to the training data	-Data -Configuration	-Function

Stage	Operation	Description	Input	Output
	Model Evaluation	Apply evaluation strategies to identify and understand the quality and performance of the model	-Model -Configuration -Data	-Report
	Model Validation	Validate the quality of the model to decide if the model can be deployed	-Report -Configuration	-Status
	Model and AI product documentation	Generate human readable documentation in order to facilitate knowledge transfer and increase transparency.	-Reports	-Documentation
Operationalization	Model Deployment	Serve model outputs to the end users. It acts as a wrapper to the model artifacts and relevant parameters	-Function -Configuration	-Service
	Model Monitoring	Implement a function to deliver model outputs to the end users. It wraps the model artifacts and relevant parameters	-Service -Data	-Logs
	Production Data Monitoring	Procedure that monitors production data and computes metrics based on data characteristics	-Data	-Reports
	System Monitoring	This implements application-level monitoring of the deployed service itself, such as a Flask web service or Kubernetes-hosted service.	-Logs	-Reports
	Pre-inference Data Transformations	Adjustments made to production data prior to feeding it into the model	-Data	-Data
	Post-inference Data Transformations	Adjustments made to the model's output to prevent data privacy breaches or to filter information.	-Data	-Data

3.2. AI system requirements

The second axis of the framework captures all requirements that an AI-enabled solution must satisfy, driven by the specific needs of the public sector [31, 6]. Designed to be modular, the framework allows the integration of additional dimensions of requirements, enabling the incorporation of diverse features and implementation tools. These elements are systematically mapped onto the horizontal operations of AI development lifecycle.

We have delineated six fundamental requirements that a basic AI solution must satisfy and facilitate for implementation. These represent the minimal core criteria for evaluating not only the model itself but also key performance indicators (KPIs) such as training data quality, inference response time or number of resources utilized. We recommend including aspects such as quality, fairness, optimization, adaptability, transparency, and privacy, as these are among the most relevant considerations highlighted in the OECD's report on the use of AI applications by governments [31]. To minimize the overhead that the framework's comprehensiveness may impose on simple or small-scale products, it is advisable to focus on a single requirement dimension or to restrict the operation categories to only those that are essential.

Quality. Among these aspects, quality encompasses data integrity, model performance, system and pipeline efficiency. A critical step is assessing the available data, including its volume, quality, and formats, to determine viable machine learning approaches. This analysis also helps stakeholders decide whether additional data collection is necessary to achieve the desired model performance or data augmentation techniques are convenient to apply. **Fairness**. The second aspect under consideration remains one of the most critically important topics in the field of machine learning. Within the research community, discussions typically center on two key aspects: (1) quantifying a model's fairness and (2) implementing interventions, during data pre-processing, model training [32], or post-processing, to improve fairness metrics [33]. Different philosophical interpretations of fairness, such as equality vs. equity are operationalized through distinct fairness measures and mitigation operations. **Optimization**. To minimize time and resource utilization during both the training and deployment phases, it is essential to integrate optimization strategies into the design of AI systems. This approach enables the systematic incorporation of performance-aware management principles throughout the machine learning lifecycle, ensuring alignment with predefined performance metrics and operational efficiency standards. **Adaptability**. Another important consideration is the adaptability of the AI system when it is reused by other public organizations seeking to reproduce it for similar use cases. **Privacy**. Among the various domains of responsible engineering, privacy has seen the most significant regulatory developments in recent years, with numerous jurisdictions enacting dedicated legislation. **Transparency**. An equally significant factor when developing AI-based solutions is the traceability and the communication of descriptive and prescriptive aspects that are relevant to different AI stakeholders [34].

3.3. The visual representation of the framework

The design of the proposed framework, is shown in Figure 2. The framework serves as an integrator of existing open-source technical tools and libraries. It organizes implementation procedures, denoted as <placeholder_procedure>, into a matrix considering both the AI development stages and the envisioned requirements. Earlier, we introduced the components that define the horizontal and vertical dimensions that coordinate the end-to-end development and deployment of an AI system.

The framework has a dual approach towards AI implementation and validation, which makes it accessible to both AI developers and compliance experts. On the one hand, it helps to explicitly and transparently declare and implement all requirements by leveraging existing tools and libraries. On the other hand, by embedding requirements into every operation of the AI lifecycle, our framework ensures that quality assurance, transparency, fairness and other aspects are built-in and evidence-based attributes of AI systems, rather than an afterthought. Moreover, it addresses the limited reusability of AI solutions in the public sector by responsibly engineering them. Finally, the framework plays a crucial role in implementing requirements by providing a systematic and structured approach to ensure that AI systems are developed ethically and comply with established service level agreements.

4. Filling in the framework

The following proof-of-concept example demonstrates how the framework can be instantiated with tools and modular implementations aligned with both specific requirements and AI development pipeline. We assume that each tool employed to address specific requirements is used to carry out a particular operation, with its inputs and outputs explicitly defined. This way, the operations create an abstraction layer that provide a level of uniformity to the employed toolkits. Additionally, the framework can be

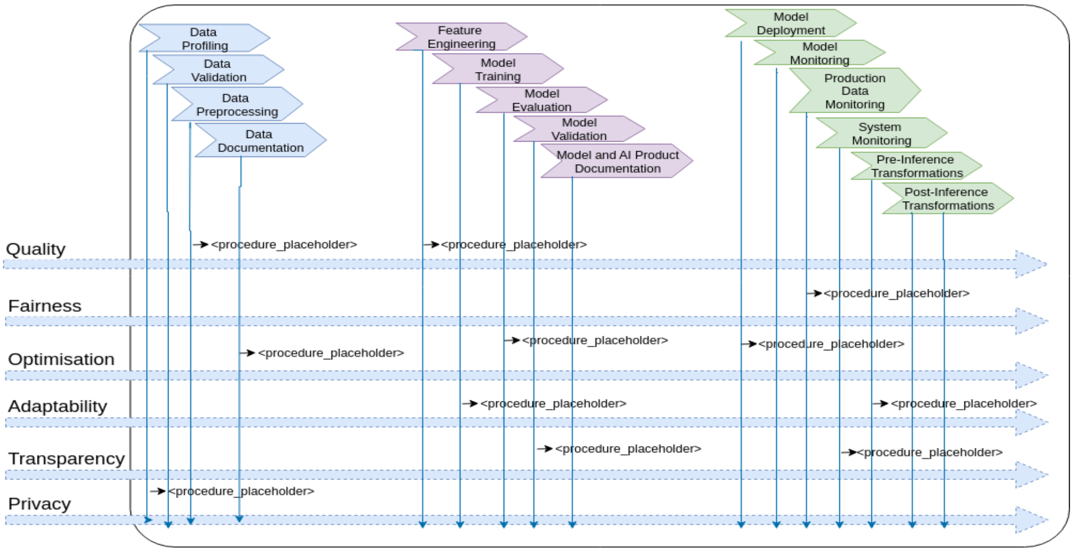

Figure 2: A visual representation of the enhanced AI lifecycle framework, which presents a multifaceted strategy composed of: (a) a horizontal axis representing the three AI lifecycle stages and their corresponding 15 operations, and (b) a vertical axis representing the requirements dimensions that the AI system must satisfy. The framework organizes the implementation procedures, referred to as `<placeholder_procedure>`, into a matrix that aligns AI development stages with the corresponding requirements. Each `<placeholder_procedure>` encapsulates implementation code that leverages a specific library to address the requirement on the vertical axis and is executed during each specific operation of the AI product lifecycle. The arrows indicate the operation to which each procedure belongs and the associated requirement aspect. Blue indicates operations belonging to the Data Preparation stage, purple corresponds to the Modeling stage, and green represents the operations in the Operationalisation stage.

implemented alongside an existing cloud-native MLOps or AI platform, such as Google Vertex AI [16] or DigitalHub AI platform [17]. To enable such integration, the implementation procedures must be adapted to comply with the governance and tool orchestration mechanisms required by the target platform.

4.1. Example: Classification model trained on tabular data

This use case presents a classification model designed to identify the most suitable candidates for hiring a new employee [35]. Figure 3 illustrates the populated framework. First, we focus on defining the requirement dimensions that the AI-enabled solution should meet. Specifically, the stakeholders define the list of the principles that the AI product should comply with based on the output of the evaluation reports [5] during the design phase. For this use case the dimensions of the requirements that are relevant are: quality, fairness, adaptability, transparency and privacy. Second, we select the proper technical tools that address each requirement and associate them to the specific AI lifecycle operations following the semantic we defined in Section 3.1. For example, for addressing quality aspects we may use the implementation procedures that use frictionless library and are both executed during data validation operation. Next, to address fairness requirements we utilize AI Fairness 360 library. The procedures named compute_dataset_fairness_metrics and transform_to_mitigate_bias are executed during data preprocessing operation, whereas during feature engineering is used calculate_feature_importance. Instead, train_fairness_tailored is executed as a fairness-tailored training procedure for the model. The other procedures that evaluate the model regarding fairness metrics are: compute_model_fairness_metrics and check_fairness_metrics. For the adaptability dimension, we utilize the Evidently library, which offers utilities for detecting data and concept drift. In terms of deployment, the model_deploy procedure implements a service function that instructs the hosting AI platform to leverage the KServe engine for scalable model serving.

[16] https://cloud.google.com/vertex-ai?hl=en
[17] https://scc-digitalhub.github.io/docs/

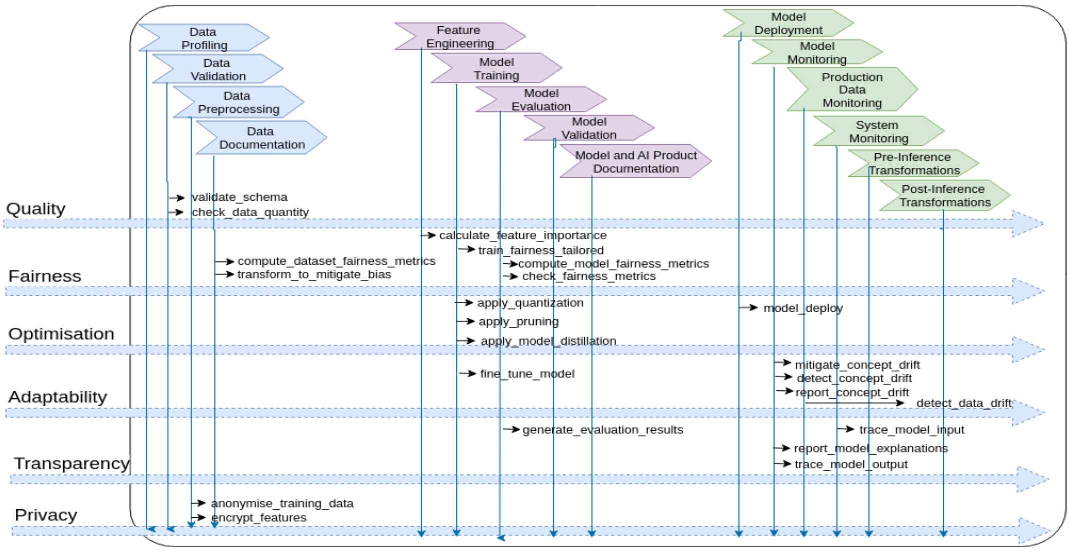

Figure 3: Enhanced structured AI lifecycle framework - proof-of-concept for classification model trained on tabular data for a hiring system

5. Conclusion

In this paper, we described the necessary operations that explicitly operationalize AI system requirements, contributing to implement responsible, modular, and reusable AI systems. Researchers and industry professionals in AI have been actively working on developing tools and methods that aid in transitioning experimental models to production deployment. By addressing the gaps in current frameworks that bring together procedural and technical approaches, our proposed enhanced lifecycle effectively facilitates the alignment between the implementation and the specific requirements of the AI system. Although existing libraries provide solutions for components of AI systems, they often fail to clearly specify the requirement aspect they address. Some approaches only partially cover some aspects, lacking a holistic perspective on the overall AI system. The clear structure of our framework explicitly connects each particular implementation to its relevant operation category and the specific requirement aspect it addresses. Moreover, the framework has a dual approach towards AI development lifecycle and requirements characteristics, enabling automation of AI product development and also verification. The framework contributes to the reuse of AI products by offering a systematic methodology for developing them by transparently and explicitly describing and assembling the components of an AI system. Although the framework enables non-technical AI specialists to adopt AI easily and responsibly, especially in areas with low level of AI adoption like the public sector, a potential limitation is the necessity to establish procedures for conflicting requirement dimensions, such as balancing accuracy with fairness. This issue can be addressed earlier in the AI product's design phase through a proactive approach, which aids in resolving ethical dilemmas in AI and also guides the implementation and evaluation phases.

6. Acknowledgments

The authors acknowledge the support of the “AIxPA - Artificial Intelligence in the Public Administration system” area of Flagship Project - Progetto Bandiera PNC-A.1.3 “Digitalizzazione della pubblica amministrazione della Provincia Autonoma di Trento”.

7. Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT and Paperpal in order to: Grammar and spelling check, paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] 2023 OECD Digital Government Index: Results and key findings. OECD Public Governance Policy Papers, no. 44, OECD Publishing, Paris (2024). URL: https://doi.org/10.1787/1a89ed5e-en.

[2] L. Tangi, C. Van Noordt, A. P. Rodriguez Müller, The challenges of AI implementation in the public sector. an in-depth case studies analysis, in: Proceedings of the 24th Annual International Conference on Digital Government Research, ACM, 2023, pp. 414–422. URL: https://dl.acm.org/doi/10.1145/3598469.3598516. doi:10.1145/3598469.3598516.

[3] H.-L. E. G. on Artificial Intelligence (AI HLEG), Ethics guidelines for trustworthy ai; european commission: Brussels, belgium (2019). URL: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai.

[4] The Responsible Machine Learning Principles (2024). URL: https://ethical.institute/principles.html#commitment-1.

[5] R. V. Zicari, J. Brodersen, J. Brusseau, B. Düdder, T. Eichhorn, T. Ivanov, G. Kararigas, P. Kringen, M. McCullough, F. Möslein, N. Mushtaq, G. Roig, N. Stürtz, K. Tolle, J. J. Tithi, I. van Halem, M. Westerlund, Z-inspection®: A process to assess trustworthy AI 2 (2021). URL: https://ieeexplore.ieee.org/document/9380498/. doi:10.1109/TTS.2021.3066209.

[6] OECD AI Principles Overview, AI system lifecycle (2024). URL: https://oecd.ai/en/ai-principles.

[7] J. Pineau, P. Vincent-Lamarre, K. Sinha, V. Lariviere, A. Beygelzimer, Improving reproducibility in machine learning research (2021).

[8] D. Kreuzberger, N. Kühl, S. Hirschl, Machine learning operations (MLOps): Overview, definition, and architecture 11 (2023). URL: https://ieeexplore.ieee.org/document/10081336/. doi:10.1109/ACCESS.2023.3262138.

[9] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, D. Dennison, Hidden technical debt in machine learning systems, in: Advances in Neural Information Processing Systems, volume 28, Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper_files/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html.

[10] Tools for trustworthy AI: A framework to compare implementation tools for trustworthy AI systems, 2021. URL: https://www.oecd.org/en/publications/tools-for-trustworthy-ai_008232ec-en.html. doi:10.1787/008232ec-en, series: OECD Digital Economy Papers Volume: 312.

[11] Catalogue of Tools Metrics for Trustworthy AI, 2024. URL: https://oecd.ai/en/catalogue/tools?approachIds=1&page=1&lifecycleIds=2&lifecycleIds=1.

[12] M. T. M. A. J. M. J. W. Y. Floridi, L.; Holweg, capai—A Procedure for Conducting Conformity Assessment of AI Systems in line with the EU Artificial Intelligence Act, 2022. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4064091.

[13] M. Fikardos, K. Lepenioti, D. Apostolou, G. Mentzas, Trustworthiness optimisation process: A methodology for assessing and enhancing trust in AI systems 14 (2025). URL: https://www.mdpi.com/2079-9292/14/7/1454. doi:10.3390/electronics14071454.

[14] M. T. Baldassarre, D. Gigante, M. Kalinowski, A. Ragone, Polaris: A framework to guide the development of trustworthy ai systems, in: Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI, CAIN '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 200–210. URL: https://doi.org/10.1145/3644815.3644947. doi:10.1145/3644815.3644947.

[15] V. Vakkuri, K.-K. Kemell, M. Jantunen, E. Halme, P. Abrahamsson, Eccola — a method for implementing ethically aligned ai systems, Journal of Systems and Software 182 (2021) 111067. URL:

https://www.sciencedirect.com/science/article/pii/S0164121221001643. doi:https://doi.org/10.1016/j.jss.2021.111067.

[16] J. Baker-Brunnbauer, TAII Framework. In Trustworthy Artificial Intelligence Implementation: Introduction to the TAII Framework, 2022.

[17] G. Stettinger, P. Weissensteiner, S. Khastgir, Trustworthiness Assurance Assessment for High-Risk AI-Based Systems, IEEE Access 12 (2024) 22718–22745. doi:10.1109/ACCESS.2024.3364387.

[18] B. Li, P. Qi, B. Liu, S. Di, J. Liu, J. Pei, J. Yi, B. Zhou, Trustworthy AI: From principles to practices 55 (2023) 1–46. URL: https://dl.acm.org/doi/10.1145/3555803. doi:10.1145/3555803.

[19] V. S. Barletta, D. Caivano, D. Gigante, A. Ragone, A rapid review of responsible ai frameworks: How to guide the development of ethical ai, in: Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering, EASE '23, ACM, 2023, p. 358–367. URL: http://dx.doi.org/10.1145/3593434.3593478. doi:10.1145/3593434.3593478.

[20] OECD framework for the classification of AI systems (2022).

[21] E. U. A. for Cybersecurity, M. Adamczyk, N. Polemi, I. Praça, K. Moulinos, A multilayer framework for good cybersecurity practices for AI – : security and resilience for smart health services and infrastructures, European Union Agency for Cybersecurity, 2023. doi:doi/10.2824/588830.

[22] A. Singla, Machine learning operations (MLOps): Challenges and strategies 2 (2023) 333–340. URL: https://jklst.org/index.php/home/article/view/107. doi:10.60087/jklst.vol2.n3.p340.

[23] M. Testi, M. Ballabio, E. Frontoni, G. Iannello, S. Moccia, P. Soda, G. Vessio, MLOps: A taxonomy and a methodology 10 (2022) 63606–63618. URL: https://ieeexplore.ieee.org/document/9792270/;jsessionid=84941428F3357304322A00024BA33EE0. doi:10.1109/ACCESS.2022.3181730.

[24] M. Arnold, J. Boston, M. Desmond, E. Duesterwald, B. Elder, A. Murthi, J. Navratil, D. Reimer, Towards Automating the AI operations lifecycle, 2020. URL: http://arxiv.org/abs/2003.12808. doi:10.48550/arXiv.2003.12808.

[25] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, T. Gebru, Model cards for model reporting, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, ACM, 2019. URL: https://doi.org/10.1145%2F3287560.3287596. doi:10.1145/3287560.3287596.

[26] M. Pushkarna, A. Zaldivar, O. Kjartansson, Data cards: Purposeful and transparent dataset documentation for responsible ai, 2022. arXiv:2204.01075.

[27] L. Derczynski, H. R. Kirk, V. Balachandran, S. Kumar, Y. Tsvetkov, M. R. Leiser, S. Mohammad, Assessing Language Model Deployment with Risk Cards, 2023. URL: http://arxiv.org/abs/2303.18190, arXiv:2303.18190 [cs].

[28] A. Celepija, A. Palmero Aprosio, B. Lepri, R. Kazhamiakin, AI product cards: a framework for code-bound formal documentation cards in the public administration, Data 38; Policy 7 (2025) e1. doi:10.1017/dap.2024.55.

[29] D. Golpayegani, I. Hupont, C. Panigutti, H. J. Pandit, S. Schade, D. O'Sullivan, D. Lewis, AI cards: Towards an applied framework for machine-readable AI and risk documentation inspired by the EU AI act, 2025. URL: http://arxiv.org/abs/2406.18211. doi:10.48550/arXiv.2406.18211. arXiv:2406.18211 [cs], version: 1.

[30] Understanding and Managing the AI lifecycle. AI Guide for Government (2023). URL: https://coe.gsa.gov/coe/ai-guide-for-government/understanding-managing-ai-lifecycle/.

[31] OECD, Governing with artificial intelligence: Are governments ready? (2024). doi:10.1787/26324bc2-en.

[32] M. Magnini, G. Ciatto, R. Calegari, A. Omicini, Enforcing fairness via constraint injection with FaUCI (2024).

[33] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. N. Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, Y. Zhang, AI fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias, 2018. URL: http://arxiv.org/abs/1810.01943.

[34] F. Königstorfer, S. Thalmann, AI Documentation: A path to accountability, Journal of Responsible Technology 11 (2022) 100043. URL: https://linkinghub.elsevier.com/retrieve/pii/S2666659622000208.

doi:10.1016/j.jrt.2022.100043.

[35] C. Calluso, M. G. Devetag, Discrimination in the hiring process – state of the art and implications for policymakers, Equality, Diversity and Inclusion: An International Journal 43 (2024) 103–121. URL: http://www.emerald.com/edi/article/43/9/103-121/1214508. doi:10.1108/EDI-10-2023-0340.