

Towards a Process Reference Model for Machine Learning Applications: Challenges and Opportunities

Antonio Crespi^{1,†}, Maria Monserrat^{1,†}, Antonia Mas^{1,†} and Antoni-Lluís Mesquida^{1,*,†}

¹Universitat de les Illes Balears (UIB), Cra. De Valldemossa, km 7.5, 07122 Palma, Spain

Abstract

The development of machine learning applications is increasingly subject to quality, compliance, and maintainability demands that exceed the capabilities of current workflow-oriented approaches. While numerous lifecycle models for machine learning have been proposed, they lack the formal structure required to support process assessment, standardization, or maturity evaluation. In contrast, software engineering has long benefited from process reference models grounded in international standards such as ISO/IEC/IEEE 24774 and ISO/IEC 33004. This paper presents the initial foundation for a process reference model for machine learning, developed after a systematic analysis of existing machine learning lifecycle models and aligned with ISO-style process specification practices. The approach formalizes machine learning processes in terms of purpose, inputs, outputs, and outcomes, and supports their eventual use in capability and maturity frameworks.

Keywords

Machine Learning, Artificial Intelligence, Lifecycle Models, ISO Standards, Process Reference Model

1. Introduction

Unlike traditional software engineering, where process-oriented development is supported by decades of standardization efforts, machine learning (ML) development remains largely unstructured. Most current practices rely on high-level workflows or tool-driven pipelines that offer limited support for systematic process definition, assessment, or improvement. Although numerous lifecycle models for ML have been proposed [1], they typically lack formal specifications of process purpose, inputs, outputs, and outcomes. This impedes reproducibility, complicates quality assurance, and hinders alignment with compliance requirements in regulated domains.

Efforts to address this gap have emerged from both academic and industrial initiatives. Frameworks such as CRISP-DM [2] and TDSP [3] have been widely used to organize ML workflows, and recent standards such as ISO/IEC 5338 [4] represent important steps toward lifecycle-oriented guidance for Artificial Intelligence (AI) systems. Nonetheless, these models stop short of providing the type of process reference model (PRM) needed to support structured development, cross-organizational alignment, or process capability assessment. Existing ML lifecycle models are diverse in terminology, inconsistent in structure, and not readily compatible with established software engineering standards.

This paper presents the initial foundation for a PRM tailored to the development of ML applications. The proposed approach draws on ISO/IEC 33004 [5] and ISO/IEC/IEEE 24774 [6], which define the structure and attributes required for process reference and assessment models. By reinterpreting lifecycle models identified in a prior systematic literature review [1] through the lens of ISO-style process specification, this work defines a methodology to formalize ML processes in a consistent and evaluable manner. The aim is not to create a new lifecycle model, but to provide a structured and standard-aligned basis for process modeling, assessment, and future maturity modeling in ML engineering.

ECAI'25: Workshop on Machine Learning Operations (MLOps25), October 25, 2025, Bologna, IT

*Corresponding author.

[†]These authors contributed equally.

✉ antonio.crespi@uib.eu (A. Crespi); maria.monserrat@uib.cat (M. Monserrat); antonia.mas@uib.es (A. Mas); antoni.mesquida@uib.es (A. Mesquida)

ORCID 0009-0000-0046-8330 (A. Crespi); 0000-0002-7485-4709 (M. Monserrat); 0000-0003-4188-8088 (A. Mas); 0000-0002-1191-6220 (A. Mesquida)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The paper is structured as follows. Section 2 reviews the background and related work, including existing ML lifecycle models, the role of standardization in software and systems engineering, and recent efforts to structure ML development through international standards. Section 3 outlines gaps and opportunities for a standardized PRM. Section 4 introduces the methodology used to define the PRM. Section 5 discusses the rationale behind the model's structure, its alignment with the literature, and illustrates how a specific process would be described. Finally, section 6 summarizes the main contributions of the paper and outlines the planned phases of future work.

2. Background and Related Work

This section provides an overview of prior work relevant to the definition and structuring of processes in ML development. It begins with a summary of existing ML lifecycle models, drawing on a systematic literature review that catalogues and synthesizes their phases and activities. It then introduces the concept of process models as used in software engineering, describing their structure and function within established lifecycle standards. Finally, it outlines the main international standardization efforts in ML and AI, focusing on the most prominent ISO/IEC initiatives and their contributions to formalizing processes in this domain.

2.1. ML Lifecycle Models

The systematic literature review conducted in [1] identified and analyzed 18 primary studies, from which 14 distinct ML lifecycle models and 4 sub-models were extracted. These models vary in structure and scope, but collectively reflect increasing efforts to address the specific demands of ML development.

A key finding of the review is the considerable variability in how lifecycle models structure and define development processes. Some models are narrowly scoped, focusing primarily on model development and deployment, while others attempt to address broader concerns such as compliance, collaboration, or long-term monitoring. However, few models provide an integrated view that spans the full range of activities—from early-stage planning and data governance to post-deployment adaptation. Notably, activities related to model updating, ethical considerations, and organisational roles are underrepresented across the literature. This heterogeneity suggests a lack of convergence on the necessary components of a complete ML lifecycle and highlights the absence of a shared conceptual foundation.

The review also categorised lifecycle activities into four broad groups: (1) objective and scope definition, (2) data, (3) model, and (4) operation activities. While some categories, such as data preparation and model training, are frequently addressed, others—such as model monitoring, project review, or risk analysis—appear only in a subset of models. These inconsistencies indicate that current ML lifecycle models tend to emphasize specific concerns (e.g., technical implementation or data workflows) while neglecting others (e.g., governance, transparency, or traceability). This fragmented coverage complicates comparisons between models and limits their applicability as general-purpose development frameworks.

2.2. Process Models in Software Engineering

Process models in software engineering provide structured frameworks that define how software is developed, validated, deployed, and maintained. Their main objective is to support systematic development by describing the activities, roles, artefacts, and relationships involved in producing and evolving software systems. These models serve as a foundation for planning, execution, monitoring, and improvement, helping teams achieve consistent results, maintain traceability, and ensure quality across the software lifecycle.

In the literature and in practice, the term lifecycle model often refers to high-level development strategies that organize the overall sequence of phases in a project, such as analysis, design, implementation, and maintenance. Common lifecycle models include the Waterfall model [7], which follows a linear and sequential flow; the Spiral model [8], which incorporates iterative development with risk management; and the V-model [9], which emphasizes parallel validation and verification for each development phase.

Later approaches such as incremental and iterative models, Agile methodologies [10], and DevOps practices [11] have placed more emphasis on adaptability, collaboration, and automation.

These lifecycle models are often supported or operationalized through process models, which describe in more detail the individual processes that occur within each phase. A process model defines what each process is intended to achieve (its purpose), what outcomes should be produced, and what activities, inputs, and outputs are involved. Process models provide the structure needed to implement a lifecycle model consistently, and they allow teams to tailor practices, assign responsibilities, and measure performance.

The ISO/IEC/IEEE 12207 [12] standard is a prominent example of a process model framework. It defines a comprehensive set of processes for software lifecycle management, including primary (development), supporting (verification, validation), and organizational (management, improvement) processes. ISO/IEC/IEEE 12207 is designed to be compatible with various lifecycle models and can be used in combination with ISO/IEC/IEEE 15288 [13] for systems engineering. It has been widely adopted in both industry and public sectors and serves as a basis for software process assessments and certifications.

Process models contribute to software development in several important ways. First, they support standardization by providing a shared terminology and structure for organizing development activities. This reduces ambiguity and improves communication within teams and across organizational boundaries. It also facilitates training, knowledge transfer, and integration with external stakeholders.

Second, process models promote repeatability and traceability. When activities are clearly defined, it becomes easier to execute them consistently across projects. Artefacts, decisions, and responsibilities can be tracked more effectively, which is particularly important in large or long-lived systems where documentation and auditability are necessary.

A third key contribution is to quality assurance. Many models, such as the V-Model and ISO/IEC/IEEE 12207, explicitly include verification and validation steps within the process structure. This supports early detection of defects and helps ensure that the final software meets its intended requirements. In regulated domains, such integration is often essential for demonstrating compliance with safety or reliability standards.

Process models also enable process improvement. When the processes are well-defined, they can be assessed and improved using formal frameworks such as the Capability Maturity Model Integration (CMMI) [14] or the ISO/IEC 33000 series. These assessments help organizations identify weaknesses, benchmark their practices, and establish goals for continuous improvement.

Finally, in many domains, structured processes are a prerequisite for regulatory compliance. Sectors such as aerospace, automotive, finance, and healthcare require formal lifecycle documentation and traceability to meet legal or certification requirements. In these contexts, adopting a recognized process model is not only a best practice but often a necessity.

2.3. ISO Standards for ML and AI

Recent efforts to formalize the development of ML and AI systems have led to the emergence of several ISO/IEC standards specifically addressing lifecycle processes, data quality, and governance in AI. These standards are primarily developed under the ISO/IEC JTC 1/SC 42 subcommittee, which focuses on AI-specific standardization.

A central contribution is ISO/IEC 5338, which defines lifecycle processes for AI systems and explicitly adapts lifecycle models such as those in ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288 to AI-specific contexts. ISO/IEC 5338 introduces novel processes such as AI data engineering, continuous validation, and iterative model updates, accounting for the dynamic nature of ML systems. While it follows the structure of software lifecycle standards, it also incorporates AI-specific characteristics such as data dependencies, probabilistic behavior, and retraining.

Other standards address supporting concerns. ISO/IEC 5259-1 [15] focuses on data quality for analytics and ML, establishing terminology and practices to ensure reliable datasets. ISO/IEC 22989 [16]

and ISO/IEC 23053 [17] contribute to terminological consistency and provide high-level conceptual frameworks for describing AI systems using ML.

Governance and risk management are addressed by ISO/IEC 42001 [18], which defines a management system for AI and provides organizational processes to ensure compliance, accountability, and ethical oversight. It aligns structurally with ISO 9001 [19], but introduces AI-specific provisions such as risk and impact assessment, lifecycle documentation, and fairness assurance. Yet, like ISO/IEC 5338, it does not offer detailed guidance for change control, stakeholder responsibility, or lifecycle traceability.

3. Gaps and Opportunities

While the previous section described recent advances in ML lifecycle models and the development of ISO standards for AI, important limitations remain. Even though there is growing interest [20] in formalizing and standardizing ML development, current models and standards still have several problems that reduce their usefulness in practice. This section summarizes the main issues found in existing ML lifecycles and in the ISO standards that aim to support ML and AI systems.

Regarding lifecycle models, many have a limited scope. They often focus on model development and deployment, but give little attention to earlier stages like project planning or later stages like monitoring and maintenance. As a result, important aspects such as traceability, documentation, and compliance with regulations are not well supported [21][22]. Even when feedback loops or post-deployment steps are included, they are usually described in general terms and not as clear, structured processes. In addition, the terminology and grouping of activities vary a lot across models, which makes comparison difficult and prevents a shared understanding of the lifecycle.

Post-deployment activities are another weak point [23]. Only a few models clearly describe how to monitor, update, or retrain models once they are deployed. These steps are needed to effectively deal with problems like model drift or performance drop over time. Ethical concerns such as fairness, transparency, and explainability are also not well represented [24]. They are sometimes mentioned, but rarely included as formal activities in the lifecycle. Furthermore, many models do not consider the different roles in ML teams, such as MLOps engineers or data engineers, which leads to an incomplete view of how real ML projects are organized and managed.

Similar gaps can be found in the current ISO/IEC standards for ML and AI. Standards like ISO/IEC 5338 and ISO/IEC 42001 introduce useful structures for lifecycle management, and governance, but they are not yet as detailed or mature as traditional software engineering standards like ISO/IEC/IEEE 12207 or the ISO/IEC 33000 series. In particular, there is no standard that defines how to assess or improve the maturity of ML development processes. This makes it difficult for organizations to evaluate their practices or track progress over time.

There are also missing definitions in the standards. Important terms and roles used in ML development—such as federated learning, data versioning, or MLOps—are not included in current terminology standards like ISO/IEC 22989 or ISO/IEC 23053. ISO/IEC 5338 describes the iterative nature of AI development and includes processes such as data engineering and validation, but it does not clearly define how to manage changes, assign responsibilities, or ensure accountability. Similarly, ISO/IEC 42001 deals with AI-specific quality and risk issues but does not explain how to include these concerns in daily development activities. Overall, while the standardization of ML processes is improving, it still lacks the completeness and practical guidance found in established software engineering standards.

A standardized PRM for ML could help address many of these gaps by providing a clear and consistent structure for defining ML development processes. Unlike current lifecycle models, which vary in scope and terminology, a PRM would define a common set of processes, each with a clear purpose, expected inputs and outputs, and recommended practices. This would make it easier for teams to align their workflows, improve collaboration, and reduce misunderstandings about responsibilities or development steps.

A PRM could also support traceability and compliance by making sure that important activities—such as documentation, risk analysis, monitoring, and model updates—are formally included in the lifecycle.

This would help teams manage technical debt, improve reproducibility, and meet regulatory requirements, especially in high-risk domains like healthcare or finance. Including ethical and governance-related processes in the PRM would also encourage teams to treat fairness, transparency, and accountability as core development tasks, not just optional concerns.

In addition, a PRM designed according to ISO principles could support process assessment and continuous improvement. By defining capability levels or maturity indicators for each process, the PRM would allow organizations to evaluate their practices and identify areas for growth. This would fill the current gap left by the absence of ML-specific process assessment standards, and provide a foundation for more reliable, maintainable, and auditable ML systems.

4. Process Description Methodology

This section presents the methodology followed to define a PRM for ML. The approach is based on two ISO/IEC standards: ISO/IEC 33004:2015, which defines the requirements for building process reference and assessment models, and ISO/IEC/IEEE 24774:2021, which provides a specification for writing process descriptions.

The first step in creating a PRM is to define its domain and scope. According to ISO/IEC 33004, the PRM must clearly state which area it applies to. In this case, the area is development and management of ML systems. It must also explain the purpose of the model and which community of users it is intended for, such as developers, researchers, or organizations working with AI systems. If the PRM is meant to reflect a shared view among practitioners, the model should also document how that agreement was reached, or clarify if no formal consensus process was used.

Each process in the PRM must include three required elements: a name, a purpose, and a set of outcomes. These elements form the core of the model and are required to comply with ISO/IEC 33004. The process name should be short and descriptive, typically ending with the word “process”. The purpose is a high-level goal that explains why the process is needed, written in plain language beginning with “The purpose of the X process is...”. The outcomes are observable results that show when the purpose has been achieved. Each outcome must describe a concrete and positive result, such as a decision being made, a quality being verified, or a resource being delivered. These outcomes should be written clearly and be easy to assess in practice.

To support practical use of the model, we also include several optional elements recommended by ISO/IEC/IEEE 24774. These help organizations implement and tailor the model to their needs. In addition to purpose and outcomes, each process may define:

- Inputs and outputs, which describe what information or artefacts the process receives and produces.
- Base practices, which are typical actions that help achieve the outcomes.
- Roles, which identify who is responsible for carrying out the process.
- Controls and constraints, such as policies, standards, or technical limitations that apply to the process.

This structure helps teams understand what each process is for, how to perform it, and how it connects with other parts of the ML lifecycle. It also supports tailoring for different domains or organizations.

An important part of the PRM is describing the relationships between processes. ISO/IEC 33004 requires that the PRM includes a process architecture, showing how processes are related. This can include sequencing (e.g., that data preparation comes before model training), feedback loops (e.g., from monitoring back to data collection), or dependencies (e.g., that one process needs the results of another to start). This structure helps users navigate the model and adapt it to real workflows.

Although this paper does not define a complete process assessment model (PAM), the structure of the PRM is designed to support future assessment. In ISO/IEC 33004, processes must be described in a way that allows their outcomes to be used as a basis for capability or maturity evaluation. This means

that each outcome must be written clearly enough to assess whether it has been achieved, providing a solid foundation for later work on process measurement and improvement.

Finally, the methodology supports the creation of process views, as described in ISO/IEC/IEEE 24774. A process view is a customized version of the PRM for a specific context, such as a healthcare application or a regulated environment. These views reuse the same processes but may highlight or modify specific elements to suit domain-specific needs. This flexibility allows the PRM to be adapted without redefining its core structure.

In summary, this methodology combines the formal structure of ISO/IEC 33004 with the practical guidance of ISO/IEC/IEEE 24774 to define a process reference model that is both standard-compliant and useful for ML development. It provides a clear format for defining processes, supports consistent implementation, and creates the basis for future evaluation and improvement.

5. Conceptual Approach Towards a PRM for ML

This section presents the structure of the proposed PRM for ML application development. It describes how the model organizes processes into categories that align with both the findings from the systematic literature review and the methodology detailed in the previous section. The goal is to define a modular and assessable architecture that supports clarity, consistency, and future standardization. Finally, an example is provided to illustrate how an individual process would be formally defined within the PRM.

5.1. Domain and Scope

The proposed PRM is defined for the domain of ML application development. It addresses the specific needs of designing, implementing, deploying, and maintaining ML systems, including activities related to data handling, model lifecycle management, validation, monitoring, and adaptation.

The scope of this PRM includes both technical and organizational processes across the entire lifecycle of ML systems. It covers project initiation, data acquisition and processing, model development, system integration, deployment, and post-deployment activities such as monitoring, retraining, and governance. The model is technology-agnostic with respect to ML algorithms and platforms, and it can be adapted to support a broad range of application domains, including but not limited to healthcare, finance, and scientific computing.

The community of interest includes ML engineers, data scientists, software engineers, quality assurance professionals, project managers, and compliance officers engaged in the development or governance of ML systems. This proposal reflects a synthesis of insights from academic literature via a systematic review, practical lifecycle models, and process modeling standards. Although formal consensus has not yet been established, the model is grounded in established practices and documented gaps identified in the literature.

5.2. Process Categories and Architecture

Drawing on the results of the systematic literature review, the proposed PRM structures ML development into six major categories of related processes. These categories reflect the specific demands of ML workflows while aligning with the structural requirements of ISO/IEC 33004. Each category groups processes that share related concerns and lifecycle roles, building on the four high-level activity types identified in the literature review (objective and scope definition, data-related activities, model-related activities, and operation-related activities) but refining them into a more granular and assessable form. This division also facilitates alignment with process architectures from traditional software engineering standards such as ISO/IEC/IEEE 12207.

- *Initiation and Planning*: This category includes processes such as project scoping, stakeholder analysis, feasibility study, regulatory and ethical risk identification, and resource planning. These processes establish the initial conditions under which ML work is defined and executed, and

provide an early alignment between business goals, legal requirements, and technical constraints. It refines the SLR category of objective and scope definition.

- *Data Engineering*: This category includes processes for data acquisition, validation, cleaning, enrichment, versioning, and storage management. These processes aim to make data artefacts that are reliable, traceable, and reusable. Treating data engineering as a standalone category also allows for better alignment with MLOps and data-centric development practices, where data quality and control are maintained independently from modelling activities. It groups the data-related activities included in the SLR.
- *Model Development*: This category includes processes for designing, training, evaluating, and selecting machine learning models. Specific processes include model selection and configuration, hyperparameter tuning, evaluation against defined metrics, and interpretability assessment. These processes are iterative and often experimental, and this category provides the necessary structure to describe them in a reproducible and assessable way. It groups the model-related activities included in the SLR.
- *System Integration and Testing*: This category covers the processes required to integrate the model into a broader software system. It includes system-level testing, interface specification, integration testing, and verification of technical and functional requirements. Although often grouped under general “operation-related activities” in the SLR, these integration tasks require a distinct treatment in the PRM due to their role in validating the readiness of ML components for production use, especially in regulated or high-reliability environments.
- *Deployment and Operation*: This category focuses on the runtime configuration and management of ML systems. Processes in this category include environment setup, version control of models and configurations, deployment planning, performance monitoring, and incident management. It includes processes from the operation-related category in the SLR.
- *Monitoring, Adaptation, and Governance*: This category introduces processes that support the long-term maintenance and responsible use of ML systems. This contains model monitoring for drift detection, performance degradation analysis, triggering of retraining cycles, documentation of updates, compliance checks, and periodic reviews of fairness or bias. These processes are increasingly emphasized in MLOps and responsible AI frameworks, and their explicit inclusion responds to one of the main deficiencies identified in current lifecycle models.

Each category in this architecture consists of multiple processes that will be defined according to ISO/IEC 33004 and ISO/IEC/IEEE 24774. For each process, the PRM will provide a name, a purpose, and a list of outcomes that must be achieved. Where appropriate, additional elements such as inputs, outputs, base practices, and roles will be included to support the implementation and integration of the PRM. All processes will be independent but connected through a process architecture that specifies the dependencies, order, and feedback relationships between them.

Each category will contain one or more processes, and each process will be defined by a name, a purpose, and a set of outcomes, along with inputs, outputs, base practices, roles, and, where applicable, controls and constraints.

Base practices for each process will be identified using a combined standards-based and evidence-based approach. Each practice will be defined to directly support one or more of the process outcomes and to contribute to the generation of the expected outputs, supporting traceability and internal consistency. To complement this outcome-driven approach, academic literature will be reviewed to identify concrete activities described in real-world ML workflows and empirical studies. These activities will be abstracted into generalized practices and incorporated when they align with the intended outcomes and add practical value.

This architecture supports modularity and allows for tailored views of the PRM, enabling adaptation for specific organizational contexts, domains (e.g., healthcare, finance), or lifecycle models. It also enables future capability evaluations aligned with ISO/IEC 33020 [25] and maturity models aligned with ISO/IEC TS 33061 [26].

Table 1
First approach to process categorization

Category	Processes
Initiation and planning	<ul style="list-style-type: none"> • Business understanding • Feasibility study • Identify needed skillsets • Prepare testing metrics and benchmarks • Privacy and regulatory compliance • Requirement gathering • Risk analysis
Data engineering	<ul style="list-style-type: none"> • Data analysis • Data cleaning • Data collection • Data enrichment • Data integration • Data labeling • Data quality analysis • Data sources identification
Model development	<ul style="list-style-type: none"> • Model Documentation • Evaluation • Model selection • Optimization • Training • Transfer learning • Transparency and interpretability
System integration and testing	<ul style="list-style-type: none"> • Model integration
Deployment and operation	<ul style="list-style-type: none"> • Model deployment
Monitoring, adaptation and goveranance	<ul style="list-style-type: none"> • Monitoring • Project review • Updating

Table 1 presents a preliminary classification of the processes identified in the literature review, organized according to the six categories defined in this work. This classification represents an initial structuring of the activities discussed in the literature review and it is intended as a first approach to organizing the landscape of ML development processes, and both the process definitions and their categorization are expected to evolve as the processes are more thoroughly analyzed and formally defined in subsequent stages of the research.

5.3. Process Example: Data Cleaning

This section illustrates the application of the methodology described in the previous section by defining the Dta Cleaning process, one of the core processes in the Data Engineering category. The process is presented below, including its purpose, outcomes, and additional descriptive elements to support clarity, traceability, and practical use within the PRM.

Name

Data Cleaning

Purpose

The purpose of the Data Cleaning Process is to detect, correct, or remove inaccuracies and inconsistencies in raw data to ensure that datasets are suitable for training, validation, and deployment of machine learning models.

Outcomes

- Cleaned datasets are produced and documented for use in downstream modelling or evaluation processes.
- The cleaning decisions and applied transformations are recorded to support reproducibility and auditability.

Inputs

- Exploratory Data Analysis (EDA) Report to guide feature engineering, model selection, and other downstream processes (output of the Data Analysis Process).
- An integrated dataset (output of the Data Integration Process).
- A Data Quality Analysis Report documenting recommendations for improving the dataset (output of the Data Quality Analysis Process).

Outputs

- Cleaned dataset: A cleaned dataset that meets the specified quality standards.
- Data cleaning documentation: Documentation detailing the identified issues and cleaning steps, decisions, and any transformations applied to fix them.
- Data Cleaning Plan: A documented data cleaning plan detailing identified issues, selected methods, and a timeline for execution.

Roles and Responsibilities

- Data Engineer: Executes the data cleaning procedures and documents results.

Base Practices

- *BP1. Plan Data Cleaning Activities* [27]: Establish a structured plan for data cleaning by selecting appropriate techniques, tools, and strategies based on the dataset's characteristics and quality requirements. This planning should take into account the EDA Report produced by the Data Analysis Process, as well as the findings documented in the Data Quality Analysis Report.
- *BP2. Handle Missing Values* [28][29]: Address missing values in the dataset according to the data type, modelling requirements, and decisions documented in the Data Cleaning Plan. Common tasks include removing rows or columns with excessive missing values, imputing values using statistical methods such as mean, median, or mode, or applying predictive models for more complex imputations.
- *BP3. Correct Data Inconsistencies* [28]: Standardize formats, units, and categorical values to ensure internal consistency across the dataset. This activity may involve correcting inconsistent date formats, aligning units of measurement (e.g. converting all weights to kilograms), reconciling variant labels in categorical variables (e.g. "yes", "Yes", "Y"), or making sure that numerical values are within expected ranges.
- *BP4. Remove Outliers and Noise* [28][29]: Detect and handle outliers and noisy data points that may negatively affect model training. Common techniques include threshold-based filtering, domain-specific rules, or model-based approaches. Depending on the context, outliers may be removed, capped, or retained with appropriate flags.

- *BP5. Normalize and Scale Data* [28][29]: Prepare numerical features for model input by applying normalization or scaling techniques appropriate to the modelling context. Common methods include min-max normalization, standardization, and robust scaling. The choice of method should consider the distribution of the data and the requirements of downstream models (e.g. sensitivity of distance-based algorithms to feature magnitude).
- *BP6. Document Cleaning Process* [27]: Maintain a detailed log of all actions taken during the data cleaning process to support transparency, reproducibility, and auditability. The documentation should include the type of transformation applied, the affected variables or records, the rationale for each decision, and any thresholds or parameters used. This log should be linked to the Data Cleaning Plan and stored alongside the cleaned dataset to ensure that the process can be reviewed, repeated, or validated by other stakeholders.

Controls and Constraints

- Data cleaning procedures must comply with organizational data policies and regulatory requirements (e.g. GDPR).
- Cleaning operations should be non-destructive where possible.
- All actions must be reproducible and auditable, particularly for high-stakes or regulated domains.

6. Conclusion and Future Work

The PRM proposed in this work contributes to an emerging body of research that seeks to formalize the development of ML applications by drawing on principles from software engineering and international standards. Existing ML lifecycle frameworks like CRISP-DM and TDSP have been widely adopted in practice, but they lack formal process structures. These models are typically defined as high-level workflows with loosely specified activities, and they do not meet the criteria required for process reference models as defined in ISO/IEC 33004, such as explicit declarations of purpose, outcomes, and assessment-ready structure.

In contrast, the proposed PRM is aligned with the structural and terminological foundations of standards like ISO/IEC/IEEE 24774 and ISO/IEC/IEEE 12207. It formalizes ML development processes by identifying their purpose, inputs, outputs, and expected outcomes, and bridges the gap between current data science practices and the structured lifecycle management traditions of software and systems engineering. The model complements the broader goals of emerging AI standards, particularly ISO/IEC 5338, by offering a more granular and evaluable process structure that can support future assessments, governance frameworks, and compliance efforts.

However, the transition to such a structured process model presents several open challenges. One major issue is domain specificity. ML applications vary significantly across sectors in terms of their development constraints, system boundaries, and lifecycle characteristics. While the proposed PRM is designed to be domain-independent, its practical application will likely require tailoring to address regulatory requirements, risk profiles, and artifact expectations that differ across contexts. The tension between generalizability and specificity remains unresolved.

Another challenge concerns the integration of formal process models with the operational practices of MLOps. Whereas the PRM is conceptual and process-oriented, MLOps frameworks emphasize automation, monitoring, and infrastructure. Current industry tools and pipelines are typically organized around deployment and retraining workflows, and it is not yet clear how these operational stages map onto process reference models. This lack of alignment makes traceability between lifecycle-level governance and implementation-level activities more difficult, particularly in iterative or continuous delivery environments.

Furthermore, there are adoption barriers associated with industry practices and culture. Many ML teams operate with informal or ad hoc workflows driven by experimentation and rapid iteration. Adhering to a structured PRM in such contexts may be perceived as burdensome unless clear benefits can

be demonstrated. This raises questions about incentives, tooling support, and the need for intermediate representations that mediate between process models and practice.

Future work will proceed in several phases. The first phase will focus on completing the full specification of the PRM, defining all relevant ML lifecycle processes in compliance with ISO/IEC standards. In the second phase, the model will be reviewed and refined based on feedback from its application in multiple organizations and industrial domains. The third phase will explore how the PRM can be customized to suit different environmental conditions, regulatory contexts, or sector-specific requirements. Finally, the fourth phase will involve the development of a process assessment model and a process maturity model for systematic evaluation and improvement of ML development practices.

7. Declaration on Generative AI

The authors acknowledge the use of GPT-4o to assist with translation, as well as grammar and syntax improvements in this manuscript. The final content, including its structure and arguments, remains the authors' own work, with all decisions regarding terminology and interpretation made independently.

References

- [1] A. Crespi, A. Mesquida, M. Monserrat, A. Mas, Lifecycle models in machine learning development, *Expert Systems* 42 (2025). URL: <https://onlinelibrary.wiley.com/doi/10.1111/exsy.70029>. doi:10.1111/exsy.70029.
- [2] R. Wirth, J. Hipp, Crisp-dm: Towards a standard process model for data mining, in: *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, 2000.
- [3] M. Learn, Team data science process (tdsp), URL: <https://learn.microsoft.com/en-us/azure/architecture/data-science-process>.
- [4] Information technology - artificial intelligence - ai system life cycle processes, ISO/IEC 5338 (2023).
- [5] Information technology - process assessment - requirements for process reference, process assessment and maturity models, ISO/IEC 33004:2015 (????).
- [6] Systems and software engineering - life cycle management - specification for process description, ISO/IEC/IEEE 24774:2021 (????).
- [7] W. W. Royce, Managing the development of large software systems, *Proceedings of IEEE WESCON* (1970) 328–388.
- [8] B. Boehm, W. J. Hansen, Spiral Development: Experience, Principles, and Refinements Spiral Development Workshop COTS-Based Systems, Technical Report, 2000. URL: <http://www.sei.cmu.edu/publications/pubweb.html>.
- [9] K. Forsberg, H. Mooz, The relationship of system engineering to the project cycle, *INCOSE International Symposium* 1 (1991) 57–65. doi:10.1002/j.2334-5837.1991.tb01484.x.
- [10] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, D. Thomas, Manifesto for agile software development, 2001.
- [11] L. Bass, I. Weber, L. Zhu, *DevOps: a software architect's perspective*, first ed., 2015.
- [12] Systems and software engineering-software life cycle processes, ISO/IEC/IEEE 12207 (2017). doi:10.1109/IEEESTD.2017.8100771.
- [13] Systems and software engineering - system life cycle processes, ISO/IEC/IEEE 15288 (????).
- [14] M. B. Chrissis, M. Konrad, S. Shrum, *CMMI for Development: Guidelines for Process Integration and Product Improvement*, third ed., 2011.
- [15] Artificial intelligence - data quality for analytics and machine learning (ml), ISO/IEC 5259-1:2024 (????).
- [16] Information technology - artificial intelligence - artificial intelligence concepts and terminology, ISO/IEC 22989 (2022).

- [17] Framework for artificial intelligence (ai) systems using machine learning (ml), ISO/IEC 23053:2022 (????).
- [18] Information technology - artificial intelligence - management system, ISO/IEC 42001 (????).
- [19] Quality management systems - requirements, ISO 9001 (????).
- [20] S. Martínez-Fernández, J. Bogner, X. Franch, M. Oriol, J. Siebert, A. Trendowicz, A. M. Vollmer, S. Wagner, Software engineering for ai-based systems: A survey (2021). URL: <http://arxiv.org/abs/2105.01984><http://dx.doi.org/10.1145/3487043>. doi:10.1145/3487043.
- [21] V. Chandrasekaran, H. Jia, A. Thudi, A. Travers, M. Yaghini, N. Papernot, Sok: Machine learning governance, arXiv (2021). URL: <http://arxiv.org/abs/2109.10870>. doi:10.48550/arXiv.2109.10870.
- [22] P. Sugimura, H. Florian, Building a reproducible machine learning pipeline, arXiv (2018). doi:10.48550/arXiv.1810.04570.
- [23] 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, Neural Information Processing Systems (2015).
- [24] R. Tong, H. Li, J. Liang, Q. Wen, Developing and deploying industry standards for artificial intelligence in education (aied): Challenges, strategies, and future directions (2024).
- [25] Information technology - process assessment - process measurement framework for assessment of process capability, ISO/IEC 33020:2019 (????).
- [26] Information technology - process assessment - process assessment model for software life cycle processes, ISO/IEC TS 33061:2021 (????).
- [27] E. Breck, S. Cai, E. Nielsen, M. Salib, D. Sculley, The ml test score: A rubric for ml production readiness and technical debt reduction, Proceedings of IEEE Big Data (2017).
- [28] P. O. Côté, A. Nikanjam, N. Ahmed, D. Humeniuk, F. Khomh, Data cleaning and machine learning: A systematic literature review, Automated Software Engineering 31 (2023). doi:10.1007/s10515-024-00453-w.
- [29] E. Rahm, H. H. Do, Data cleaning: Problems and current approaches, IEEE Data Engineering Bulletin (2000).