Computers in Industry 154 (2024) 104037

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.sciencedirect.com/journal/computers-in-industry

Fundamental requirements of a machine learning operations platform for industrial metal additive manufacturing

Mutahar Safdar [a,b], Padma Polash Paul [c], Guy Lamouche [b], Gentry Wood [d], Max Zimmermann [e], Florian Hannesen [f], Christophe Bescond [b], Priti Wanjara [b], Yaoyao Fiona Zhao [a,*]

[a] *Department of Mechanical Engineering, McGill University, Montreal, QC H3A 0C3, Canada*
[b] *National Research Council Canada, Montreal, QC H3T 1J4, Canada*
[c] *Braintoy AI, Calgary, AB T3P 0B9, Canada*
[d] *Apollo Clad Laser Cladding, a division of Apollo Machine and Welding Ltd., Leduc, AB T9E 0B3, Canada*
[e] *Faunhofer Institute for Laser Technology ILT, Aachen 52074, Germany*
[f] *BCT Steuerungs, und DV-Systeme GmbH, Dortmund 44263, Germany*

ARTICLE INFO

Keywords:
Computing infrastructure
Data analytics and machine learning
Machine learning operations platform
Fundamental and functional requirements
Industrial additive manufacturing

ABSTRACT

Metal-based Additive Manufacturing (AM) can realize fully dense metallic components and thus offers an opportunity to compete with conventional manufacturing based on the unique merits possible through layer-by-layer processing. Unsurprisingly, Machine Learning (ML) applications in AM technologies have been increasingly growing in the past several years. The trend is driven by the ability of data-driven techniques to support a range of AM concerns, including in-process monitoring and predictions. However, despite numerous ML applications being reported for different AM concerns, no framework exists to systematically manage these ML models for AM operations in the industry. Moreover, no guidance exists on fundamental requirements to realize such a cross-disciplinary platform. Working with experts in ML and AM, this work identifies the fundamental requirements to realize a Machine Learning Operations (MLOps) platform to support process-based ML models for industrial metal AM (MAM). Project-level activities are identified in terms of functional roles, processes, systems, operations, and interfaces. These components are discussed in detail and are linked with their respective requirements. In this regard, peer-reviewed references to identified requirements are made available. The requirements identified can help guide small and medium enterprises looking to implement ML solutions for AM in the industry. Challenges and opportunities for such a system are highlighted. The system can be expanded to include other lifecycle phases of metallic and non-metallic AM.

1. Introduction

Additive Manufacturing (AM) or three-dimensional (3D) printing is used to fabricate products from digital designs and has recently gained enormous attention from industry and academia (Gibson et al., 2021a). A multitude of benefits is encouraging researchers to develop AM technologies that can rival conventional manufacturing techniques at the industrial scale. Some notable advantages include tool elimination, material saving, the printing of complex designs, cost reduction, part consolidation, accelerated time-to-market, reduced storage with virtual inventory, and large-scale manufacturing of personalized products (Mehrpouya et al., 2019). AM technologies also enable specialized applications such as the printing of multifunctional or multi-material designs (Liu et al., 2021a). As a result, AM features offer solutions to both existing and new challenges faced by manufacturing enterprises and have enormous economic potential (Piller et al., 2015).

While AM offers great promise, it faces challenges that lead to quality issues in the printed parts. For example, the lengthy nature of layer-wise material addition inevitably leads to different types of defects being introduced during the process of printing parts. The common techniques for plastic-based printing are also confronted with an array of challenges. Bubbles, overfills, scars, underfill, and warpage are some of the most common defects in the material extrusion process which is a type of polymer-based 3D printing (Oleff et al., 2021). Metal AM (MAM), a subcategory of AM that can generate fully dense metallic parts, faces defects such as pores, cracks, lack of fusion, inclusions, irregularly

* Corresponding author.
E-mail address: yaoyao.zhao@mcgill.ca (Y.F. Zhao).

https://doi.org/10.1016/j.compind.2023.104037
Received 16 June 2023; Received in revised form 14 October 2023; Accepted 18 October 2023
Available online 27 October 2023

shaped features, and more (Zhu et al., 2021). This limits the potential of AM technologies to rival conventional manufacturing processes at the industrial scale. Therefore, the application of AM has not been successfully expanded to large-scale production of functional parts, especially for critical (e.g., load bearing) conditions.

The issues of AM have inspired efforts across disciplines to identify, model, and control key process-structure-property relations for quality enhancement. Therefore, AM has a very active research landscape spanning materials informatics, design theory, process and measurement science, numerical or analytical modeling, systems engineering, and development of software and hardware. Among existing solutions to AM challenges, data-driven techniques have become popular as they offer unique benefits over analytical or numerical solutions. A range of Machine Learning (ML) and Deep Learning (DL) models have been used by researchers to approach AM challenges using design, material, process, structure, and property data (Qin et al., 2022). Numerous existing reviews on data-driven AM applications highlight the different aspects of ML applications in AM, such as AM data types (Zhang et al., 2022), ML architectures (Qi et al., 2019), physics-informed ML (Mozaffar et al., 2021), in-situ process monitoring (Lin et al., 2022), non-destructive testing (Charalampous et al., 2020), and research opportunities (Qin et al., 2022).

As ML-aided AM solutions bear fruit at the laboratory scale, their subsequent adoption in the industry seems to be on the horizon. The industrial implementations of these solutions require a systems perspective to develop, deploy, manage, operate, and update the ML models of AM. While the systems engineering perspective has been considered for AM informatics before (at the data (Kim et al., 2015) and information (Mies et al., 2016; Bonnard et al., 2019) levels), no reference on managing ML models at the industrial scale exists. The majority of the efforts in AM concern standalone model deployment in open loop or simple closed-loop scenarios without a platform to manage these models (Phua et al., 2022). Similar efforts are aimed at the development of AM data management systems (Liu et al., 2020a). On the continuum of data, information, knowledge, and wisdom, the majority of the existing efforts are limited to upstream components of data and information.

Machine Learning Operations (MLOps) is a paradigm used to deploy and manage ML models at scale. MLOps refer to the set of practices recommended to develop, deploy, monitor, govern, and iterate ML models (Kreuzberger et al., 2022). An MLOps platform systematically provides these functionalities in an integrated environment. While the ML practices for production and operation can help manage ML solutions of AM, no guidelines exist to integrate the two sides for such a platform. Moreover, no work has explored systems-level perspective for managing ML models developed for AM processes. In this work, we identify fundamental roles, processes, systems, operations, and interfaces as functional components leading to practices that can be followed to develop an MLOps platform for generic AM process needs. Specifically, this work contributes to:

1. **Connecting** the status of ML-driven AM research and applications to the development of a platform for industrial deployment.
2. **Identifying** AM and ML requirements to develop an MLOps platform for industry.
3. **Detailing** functional roles, processes, systems, operations, and interfaces as fundamental components of the MLOps platform.
4. **Bridging** the knowledge gap and expertise between the domains of AM and ML to realize the cross-discipline platform.
5. **Referencing** peer-reviewed AM and ML research in support of the identified requirements.

The remainder of the paper is organized as follows. Section 2 provides a background on the status of ML-driven AM and the status of AM informatics, digitization, and systems. The gap for a systematic and integrated environment to manage ML operations is highlighted. Section 3 introduces the process and rationale to identify the requirements from associated domains leading to the components of the requirements. Section 4 and Section 5 specify the requirements for AM and ML in detail, respectively. Section 6 identifies the interfaces and their requirements in support of MLOps for industrial AM. Section 7 briefly lists the existing challenges and opportunities for MLOps in AM. Section 8 concludes the requirements and lists future works.

2. Background

This background section is divided into two parts. The first part briefly reviews the state of ML research in AM with examples from design, process, structure, and property phases. Some of the recent and representative efforts to summarize the data-driven AM research are highlighted. The perspectives drawn from these efforts hint at the real-time and in-situ application of the ML models as a key opportunity. The second part reviews research on AM informatics, digitization, and systems. These efforts are arranged into data, information, knowledge and application contexts. The challenge of managing the ML models for industrial operations highlights the existing gap in AM research and motivates the development of a dedicated ML platform to support the transition of AM specific models from proof-of-concept to production phase. In this regard, MLOps paradigm is seen as a bridge linking the ML applications and AM informatics to establish a set of practices supporting the deployment of ML models for AM operations at industrial scale.

2.1. Part I: ML in AM

The applications of ML are rapidly growing as it is being used to resolve issues at the design, process, structure, and property phases (Wang et al., 2022). At the AM design phase, structures and materials can be designed and optimized using ML. In this regard, ML applications to predict different characteristics such as design parameters (Baturynska and Martinsen, 2021), design candidacy (Zhang et al., 2021a), and design rules (Ko et al., 2021) exist. At the AM process phase, ML has been used for planning (Wang et al., 2019), monitoring (Lin et al., 2022; Zhang and Yan, 2022), controlling (Lee et al., 2021; Mahmoud et al., 2021), anomaly or defect prediction (Fu et al., 2022; Chen et al., 2021) and surrogate modeling (Roy and Wodo, 2020; Kumar et al., 2021). Process planning such as tool path (Nguyen et al., 2020), build orientation (Thomas et al., 2020), and scan strategy (Zohdi, 2019) has benefited from ML applications. In-situ monitoring can be seen as one of the major ML applications as compared to other AM lifecycle phases as depicted by numerous research efforts (Cannizzaro et al., 2021; Safdar et al., 2023a).

Open- or closed-loop control for specific AM concerns is also being researched at the process phase (Liu et al., 2020b). AM process states (Wu et al., 2019a), parameters (Mativo et al., 2018), and anomalies (Scime and Beuth, 2018) have been predicted with the help of ML models. Surrogate modeling of physical AM processes is being seen as a useful application of empirical ML solutions (Mozaffar et al., 2021). Surrogate models of process histories (Donegan et al., 2020) and process conditions (Desai and Higgs III, 2019) have been developed to bypass computationally expensive and time-consuming numerical solutions. At the AM structure phase, both macro (e.g., geometric dimensions and deviations) and micro (e.g., defects) level product characteristics have been investigated with ML (Wang et al., 2022). ML has also been applied to predict properties and performance of AM printed structures using AM data from upstream phases (Hu et al., 2021; Gaikwad et al., 2020). In most of these applications, the standalone ML models were developed and validated in the laboratory without any guidance to scale these solutions to industry.

Several systematic and comprehensive efforts have been made to summarize the growing body of scientific literature leading to challenges and opportunities being highlighted for AM development. The plethora of AM literature inspired by the strengths of data-driven approaches has resulted in several efforts to summarize its status. These can be general or specific with a focus on AM or data-driven aspects. Reviews with broader scope provide an overall status of the field with key research directions. Specific reviews, on the other hand, focus on subtopics in AM, AI, or both. The bulk of the existing surveys fall in the category of specific reviews. AM subtopics include process (Muthiah et al., 2022), material (Johnson et al., 2020), application (Lin et al., 2022), or system (Moltumyr et al., 2020). AI subtopics have focused either on the data (Wu et al., 2021) or the learners (Valizadeh and Wolff, 2022). Based on the timing of this text, over fifty surveys can be found in the literature that are directly (e.g., AI in AM) or in-directly (e.g., smart monitoring) related to data-driven AM.

Reviews by Qin et al. (2022), Tian et al. (2021) and Meng et al. (2020) are representative efforts to provide a general summary of the status of AI in AM. AI-driven smart monitoring is the most prevalent category among specific review topics. In this regard, Lin et al. (2022) and Zhu et al. (2021) have comprehensively focused on reviewing MAM condition monitoring. A key contribution in summarizing data-driven techniques in AM has been made by Johnson et al. (2020) with a specific focus on materials development. An algorithmic-equivalent of their work is the recent review on data-driven process, structure, and property modeling by Wang et al. (2022). Some researchers analyzed the existing literature from the lens of data-driven techniques and provided a summary of the most important aspects. Qi et al. (2019) reviewed Neural-Network (NN) based AM applications while Joshi et al. (2019) summarized supervision-based learning in AM. A data-centeric review was also conducted where image-based monitoring of Powder Bed Fusion (PBF) processes was thoroughly considered (Wu et al., 2021).

A common perspective presented in several review efforts points towards the in-process application of ML solutions, such as closed-loop and automated control, in-process data handling, integration of measurement and prediction modules, and deployable ML algorithms. There exist efforts to transform existing frameworks of AM informatics to meet the needs of diverse ML requirements. These efforts can be arranged into data, information, knowledge, and application levels. Moreover, smart AM paradigms have been proposed by several researchers in the context of cyber-physical systems (Al Mamun et al., 2022; Wu et al., 2019b), internet of things (Lhachemi et al., 2019), cloud computing (Wang et al., 2019) and digital twins (Phua et al., 2022). The next part arranges these efforts and links their status with the gap highlighted by the absence of an ML platform to manage end-to-end development of solutions for AM operations.

2.2. Part II: AM informatics, digitization, and systems

Research efforts on system and framework development to support AM digitalization have been organized according to their focus on data, information, knowledge, and application aspects. AM data has been increasingly growing in its diversity and magnitude prompting efforts to manage its representation, processing, storage, and use in a systematic manner. Zhang et al. (2022) reviewed the status of AM data for ML applications with a focus on data type, processing, quality, and management. They presented a simple, easy to access, and systematic platform that can be used to share AM datasets from scientific literature in support of reproducibility. In the context of system and framework development, they identified several efforts to manage AM data, some of which were linked to ML applications.

One of the representative research works with focus on AM data is on a systematic data management framework (Liu et al., 2020a), particularly a cloud-based digital twin-enabled data management framework for MAM. A similar approach on a collaborative AM data management system was proposed earlier as well (Lu et al., 2017). Their data management system aims to establish the correlations between processes, materials, and parts. Efforts on improving AM data management for better digitization are found in the literature as well. Qin et al. (2019) reviewed AM data representations highlighting their status and needs for future AM technology. Similarly, Gräßler et al. (2016) reviewed the requirement and status for AM data management with a focus on their representation. However, these works are limited to data and do not extend to the management of ML models.

AM information results from various operations on raw data in support of its subsequent usage including the diverse ML applications. Some common data operations include modeling (Bonnard et al., 2019), representation (Feng et al., 2015), packaging (Kim et al., 2017), registration (Lu et al., 2020; Feng et al., 2022), alignment (Feng et al., 2020), integration (Lu et al., 2015), fusion (Grasso et al., 2018; Yang et al., 2021), and processing, where processing can be both domain guided or generic (Safdar et al., 2023a). Lu et al. (2015) presented an integrated data schema for AM processes in order to enable better capturing, storage, and management. The core of the proposed model was based on PPR (Product, Process, and Resource), a product lifecycle management data modeling method (Lu et al., 2015). Along similar lines, the term digital thread has been coined to integrate key components of AM lifecycle data in a systematic manner (Mies et al., 2016).

Mies et al. (2016) defined digital thread as the process of capturing and analysing AM data to drive opportunities for improvement and innovation. They presented an overview of AM informatics in relation to digital thread. Bonnard et al. (2019) presented a hierarchical object-oriented model to realize AM digital thread. In addition to AM informatics, frameworks to manage specific AM operations have been proposed as well. In order to bring diverse AM process data into the same coordinate system, Feng et al. (2022) presented a multi-scale data registration framework for spatial and temporal resolution of in-situ and ex-situ AM data. Yang et al. (2021) proposed a data fusion framework to support process monitoring and control. In addition to fusion and registration of AM data, the term alignment has been used to align AM process, structure and property data with build commands (Feng et al., 2021). The frameworks to support AM information fall short of the techniques needed to manage ML models and operations.

AM knowledge, either explicit or implicit, is generated through the use of AM data and information in downstream operations that can include representation, integration, management, engineering, and transfer. Dinar and Rosen (2017) documented AM design knowledge through formal representations using Web Ontology Language (OWL). Similar to design ontologies, process ontologies have been presented to formalize AM process-related knowledge. Roh et al., (2021, 2016) proposed ontology-based process maps to identify printable zones in metal additive manufacturing. Ko et al. (2019) proposed an AM knowledge engineering framework based on ML. Park et al. (2019) proposed a framework to identify and prioritize data-driven tasks in AM through collaborative knowledge management. Xiong et al. (2020) proposed and implemented a knowledge-based computer-aided process planning tool to aid different stages of an AM process. Sanfilippo et al. (2019) proposed and implemented an ontology-based knowledge model for AM. Liang (2018) modeled knowledge of AM process planning through the use of ontologies. Kim (2019) proposed an approach to compose predictive models in metal AM to aid search of optimal process parameters. Liu et al. (2021b) proposed a framework to transfer AM knowledge among different PBF printers while keeping material and other variables same. Zhang et al. (2021b) proposed a knowledge transfer framework to support speedy process development in Aerosol Jet Printing. The existing frameworks on AM knowledge and associated systems do not extend to incorporate ML models of AM operations.

AM systems to handle specific applications also exist where models are presented as end solutions to meet specific requirements. Gribova et al. (2020) proposed a decision support software that can leverage both knowledge and past cases to support operator's decision in laser-based AM processes (Gribova et al., 2020). Ogoke and Farimani (2021) used reinforcement learning to provide closed-loop thermal control in laser PBF. Shi et al. (2020) introduced a laser beam shaping strategy to provide microstructural control of parts being printed using AM. In order to control pore formation, Khairallah et al. (2020) used synchrotron experiments and high fidelity simulation to identify the phenomenon leading to defects formation. Based on this discovery, criteria to control process variations was derived and implemented. Druzgalski et al. (2020) implemented a software package to support the rapid optimization of build geometries from initially designed models. Renken et al. (2019) implemented a closed-loop melt pool temperature control strategy using process simulation and sensors. In order to reduce the over-melting and improve build quality, Wang et al. (2020) implemented a model based feedforward control for laser power. A closed-loop high-fidelity simulation integrating finite element modeling with feedback controls was implemented to explore the parameter space for desirable part properties. Phua et al. (2022) proposed a hierarchy to categorize the nature of integration for ML models leading to a fully developed digital twin in AM. Majority of the existing AM systems in open literature focus on a single AM concern and do not extend to a platform that can support ML operations in industry.

There is a clear gap highlighted by the lack of support to deploy ML models of AM in operation. Though significant progress has been made in the direction of ML applications in AM as well as in the development of digital AM systems, no work to our knowledge exists in the open literature on an MLOps platform to manage ML models and operations of MAM at the industrial scale. Identifying the requirements for such a system is a necessary pre-requisite to develop the platform.

3. Process to identify requirements

An MLOps platform for AM brings together expertise from diverse domains. We first identify the key domains from AM and ML representing the expertise needed to realize an MLOps platform. The relevant expertise from each domain is thus described. For each identified domain, key subdomains are highlighted. These subdomains are parameterized in Section 3.2 into roles (representing skills), modules (representing systems), activities (representing processes and operations), and interactions (representing interfaces) to portray components of the requirements. The detailed description of each parameter is provided therein. These parameters from both domains are used to analyse the requirements. The analysis leads to the identification of the fundamental components of the requirements. Finally, we construct a detailed activity diagram highlighting the project level requirements of an MLOps platform in terms of the identified components.

3.1. Involved domains

The involved domains are representative of the research areas highlighted in Section 2.1 and Section 2.2. Four domains in AM are identified to have critical connection to MLOps platform, as shown in Fig. 1. These are AM Hardware Systems, AM Process Technology, AM Informatics, and Materials Science. AM Hardware Systems refers to hardware components for both regular AM tasks (e.g., energy or material supply, controller), as well as those supporting ML needs (e.g., sensors, AM data handling module). AM Process Technology refers to expertise in one of the seven standard AM technologies aimed at specific applications (ASTM, 2022). AM Informatics covers the data and information aspects of the AM side, including AM system connections, AM data handling modules, and local interfaces. Materials Science needs of AM are categorized into pre-processing (preparation of the material), processing (in-situ handling) and post-processing (characterization and evaluation) phases. The identified subdomains guide AM requirements in support of ML solutions. In this regard, generic parameters are presented to categorize them later in this subsection.

On the ML side, the process identifies Machine Learning, Data Science, and Software Engineering as the key domains to drive MLOps development, deployment, and maintenance. The ML domain covers the expertise of the model development processes. Based on the AM needs of a particular industrial system, specific model architectures can be trained to be optimal. The Data Science domain supports requirements for handling AM big data. AM systems generate huge amounts of data in diverse computer representations that has inspired concerted efforts to manage it for downstream tasks. Software Engineering is aimed at supporting the architecture development of an MLOps platform and guiding its deployment to ensure seamless operations in production.

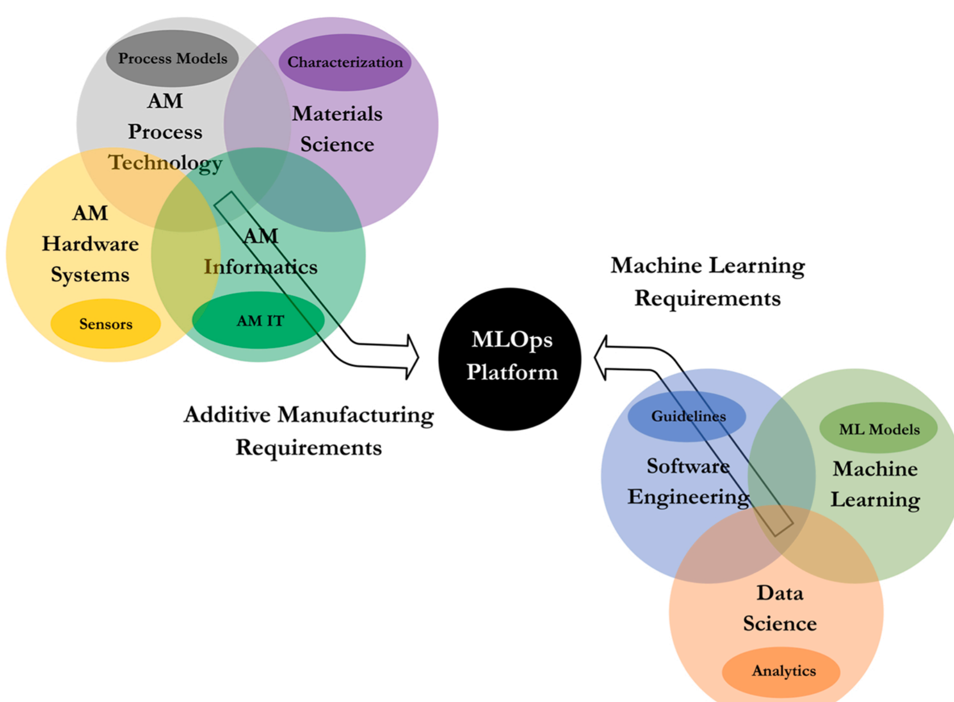

Fig. 1. ML and AM domains representing expertise needed to develop a customized machine learning platform to support industrial AM.

3.2. Analysis of requirements

In order to analyse the requirements, the needs of AM and ML subdomains are represented in terms of the required roles, modules, activities, and interactions for an MLOps platform. These terms are described below, and their specific instances are explained which are referred to throughout the paper.

Roles: Roles represent personnel with skills and experience in one or more of the subdomains from AM and ML. We use AM Actors and ML Actors to represent roles of requirements. Roles can be cross-disciplinary in nature.

Modules: Modules represent systems that provide functionalities from specific subdomains of AM and ML. On an abstract level, a module can refer to either a hardware or software system. A module can consist of several submodules. We use AM Systems and ML Systems to represent modules of requirements.

Activities: Activities represent actions from roles to develop and use the modules of an MLOps platform. In this regard, activities are divided into processes and operations. A process is used to represent development activity in AM and ML before the platform is deployed for use. Operations are used to group the set of practices from AM and ML to govern the developed modules. We use AM Processes and Operations and ML Processes and Operations to represent activities of requirements.

Interactions: Interactions represent interfaces among roles, modules, and activities associated with MLOps of AM. The interactions can take different forms (e.g., role-role, role-module, module-module). We use actor-actor, actor-system, and system-system interfaces to represent interactions of requirements.

Fig. 2 links the domain parameters identified above to the components of requirements. In this regard, the description of each numbered requirement is listed below:

Requirement#01: Expertise to develop AM process technologies.

Requirement#02: Availability of current process knowledge such as rules, models etc.

Requirement#03: Activities to develop and deploy AM process technologies.

Requirement#04: Expertise to develop AM process hardware modules.

Requirement#05: Availability of critical hardware modules of AM to support AM and ML activities.

Requirement#06: Availability of critical software modules of AM to support AM data and information.

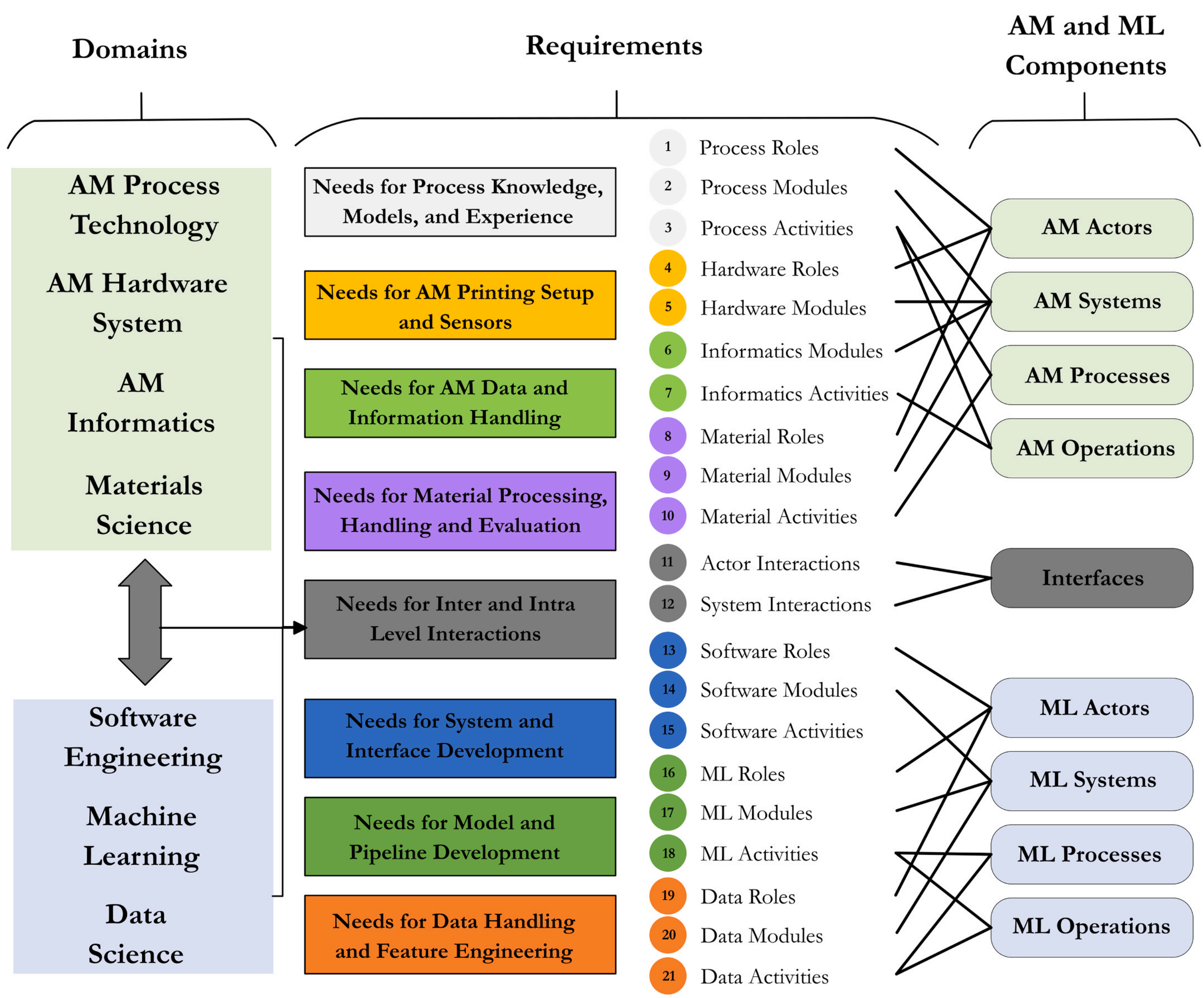

Fig. 2. Analyzing the requirements by translating involved domains and identified subdomains to specific needs. These needs for each domain are linked to the identified components of requirements.

Requirement#07: Activities to develop and deploy AM informatics modules.

Requirement#08: Expertise to develop AM materials and provide support to meet ML requirements on AM materials data and informatics.

Requirement#09: Availability of critical hardware and software modules to meet requirements on AM materials data and informatics.

Requirement#10: Activities to prepare, process, and evaluate AM materials involved in a specific process technology.

Requirement#11: Activities and modules to support the inter and intra level interactions of actors with other actors and systems.

Requiement#12: Activities and modules to support the inter and intra level interactions of systems with other systems and actors.

Requirement#13: Expertise to develop new IT solutions for AM and ML challenges.

Requirement#14: Availability of IT infrastructure to support software development.

Requirement#15: Activities to develop and deploy IT solutions.

Requirement#16: Expertise to develop ML solutions for AM needs.

Requirement#17: Availability of critical hardware and software modules to meet requirements on ML development.

Requirement#18: Activities to develop and deploy ML solutions.

Requirement#19: Expertise to develop data handling functionalities and modules.

Requirement#20: Availability of critical hardware and software modules to handle AM data for ML.

Requirement#21: Activities to develop and deploy AM data handling module.

3.3. Components of requirements

An MLOps platform, once developed, acts as an interface between the ML side and the AM side providing ML functionalities tailored to AM needs. Fig. 3 presents an activity diagram of both sides. It systematically interfaces the components of AM and ML leading to the development, deployment, and maintenance of the platform for industrial AM systems.

The **roles** represent expertise from the domains identified earlier. One person may have multiple expertise covering several of the identified roles. On the ML side, we identify ML Owner (MLO), ML Architect (MLA), Data Scientist (DS), Software Engineer (SE), Development and Operations Engineer (DevOpsE), and ML Engineer (MLE) as the required expertise. These roles support project setup, ML process and system development, ML operations and the developed MLOps platform. Their descriptions and interactions among each other are discussed in Section 5, which focuses on ML requirements. On the AM side, we identify AM Owner (AMO), AM Manager (AMM), Process Engineer (PE), Materials Engineer (ME), AM Researcher (AMR), AM Scientist (AMS) and AM Operator (AO) as the required expertise. Like ML, AM roles are involved in the process and system development in addition to project setup and continuous development. Their descriptions and interactions among each other are discussed in Section 4, which focuses on AM requirements.

The **ML Process and System Development** step can be divided into resource setup, models and methods development, and system development. The ML resources can be seen consisting of both hardware and software components. Hardware resources identified as critical include Deployment Hardware (DH), Education Hardware (EH), Training (e.g., model development) Hardware (TH), and Operations Hardware (OH), as abbreviated in Fig. 3. Software resources are divided into development and deployment resources. For development, two key ML engines or modules are related to data and models. Data Modules (DM) support data experiments dubbed as Feature Engineering (FE), while Model Modules (MM) support ML task specific functionalities, such as regression, classification, or clustering. These functionalities support model development experimentation abbreviated as ME or Model Experiments in Fig. 3. The next step of this phase represents aggregation of activities grouped under pipeline development. The term "pipeline" refers to steps of data processing and model development with the ability to automate as per the operational practices of MLOps. In this regard, two types of ML pipelines are developed, one supporting data processing and the other covering all aspects of AM-oriented ML models. These pipelines populate the ML modules and enable the platform to be deployed in support of AM operations. The details are part of Section 5, which lists ML requirements.

The **AM Process and System Development** also starts with resource setup. AM resources are divided into three groups namely AM system, AM sensors, and AM materials. AM system represents necessary components to satisfy the conditions of an AM process technology. We identify Controller System (CS), Feeding System (FS), Energy System (ES), Atmosphere System (AS), and Monitoring System (MS) as the key hardware components involved. Sensors represent added hardware components that are used to collect data of the AM process to feed ML pipelines. Based on the common sensor types in AM, these can belong to the categories of Vision Sensors (VS), Geometry Sensors (GS), or Acoustic Sensors (AS) among other types. Analogous to the development of ML pipelines, the second step of this phase is the development of AM processes based on AM knowledge that supports the execution of experiments leading to the generation of in-situ and ex-situ AM data for ML training. AM system development refers to the hardware setup on the AM side that support the interface of AM operations with ML operations. Specifically, the requirements on data generation, processing, and transmission are met on the AM end.

The **Operations Flow** on both ends highlights the set of practices that govern the ML and AM operations. At a generic scale, the ML operations start with the hardware setup to deploy an instance of the MLOps platform for development. Specific data and model modules are built later with functionalities to program and train ML solutions for AM needs. These solutions, once developed, instantiate the data and model modules paving the way for automation of pipelines. A standalone and customized version of the MLOps platform can be moved for deployment near the industrial AM system using edge computing. The AM operations follow common steps of the process lifecycle. In a typical industrial setup, this starts with the system condition validation to ensure that all parts can execute the subsequent AM operations. Depending on the AM process technique, the next step is identified as part or material setup to support the layer-wise addition of input/feedstock material. The AM system is programmed to execute a specific build in the process planning step. Once the preparations for the AM process are completed, the AM build is executed. The nature of execution depends on the level of control available that can be roughly categorized as open-loop or closed-loop system. As the process is carried out, the phenomenon of interest is monitored through installed sensors. The data from the sensors feed relevant pipelines of the MLOps.

The **Interface** represents a critical component to fulfil the requirement of implementing an MLOps platform for AM needs. It is where the operations of two systems are connected in support of ML-aided predictions. The fundamental requirement of the interface is its ability to handle incoming AM data, as well as host the automated ML pipelines of data and models for inference. In order to meet these conditions, the interface is expected to connect with the operations of both systems. In addition to system level interfaces, user interfaces and interactions among the users are also part of the project activities. We identify cloud computing and edge computing as two enabling technologies to help implement the interface of an AM tailored MLOps platform (Răileanu et al., 2018). In addition to the fundamental components identified earlier, the system also depends on auxiliary components. The auxiliary components identified on the ML side include code repositories and pipeline registries. On the AM side, data and knowledgebases are needed to store historic process records, as well as AM models and metrics. These auxiliary components from both sides enable continuous process development, a practice in both ML and AM domains.

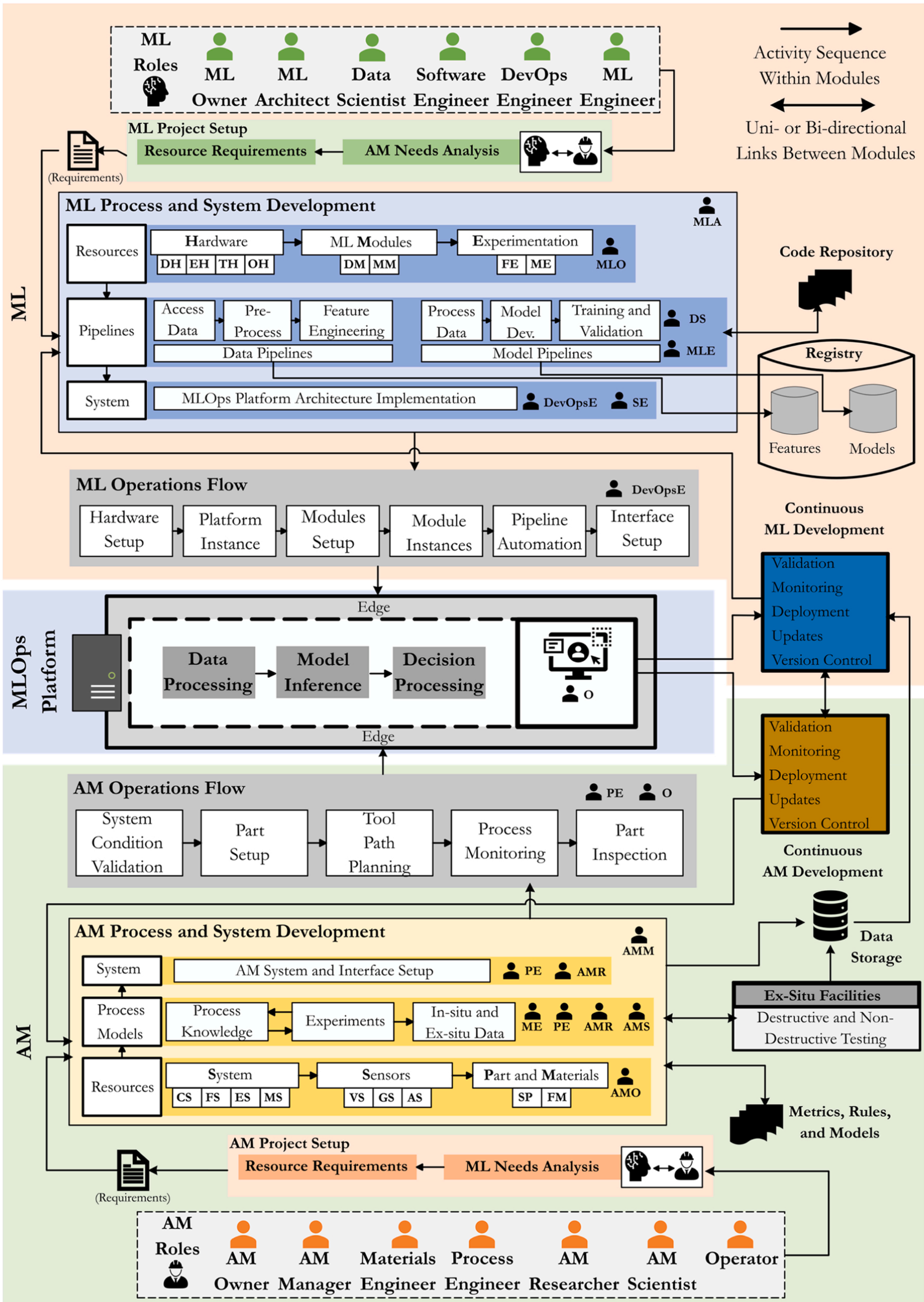

Fig. 3. This project activity diagram illustrates the complex relationship between AM and ML highlighting the various roles, processes, systems, operations, and interfaces involved. An MLOps platform (e.g., such as mlOS (BraintoyAI, 2023)) is specifically designed to align with and support the depicted operations within this context. The arrows are used to indicate the sequence of steps within each module and the link among different modules. These links can carry data, information, or decision depending on the type of modules involved in the connection.

4. Requirements for additive manufacturing

This section describes in detail the technical requirements for AM to support MLOps at the industrial scale. These requirements are decomposed into actors, process development, system development, and critical AM operations. The requirements related to the interfaces are described separately in Section 6.

4.1. AM actors

AM actors refer to the roles identified earlier. As can be seen, for MAM, a range of experts need to be involved in the development process. Fig. 4 identifies the key AM roles, describes their mandates, and highlights potential interactions among these AM actors. Table 1 summarises the required expertise of AM actors. The requirements are also enumerated to provide a concise reference for use in other tables. This is done throughout the text.

The description of each role is as follows:

AM Owner: The AM Owner identifies the business needs of AM suitable for ML applications in terms of their scale and nature. AM Owner is also responsible for making the necessary resources available to AM and ML actors per the project's requirements.

AM Manager: The AM Manager provides technical supervision to AM activities of process and system development. AM Manager also conveys the scope of AM needs for ML solutions and acts as a bridge between the AM Owner and ML actors to translate ML needs into relevant actions.

Process Engineer: The Process Engineer brings expertise in one or more of the seven standard AM technologies for specific applications and materials. Process Engineer is tasked with developing AM process to a level that can support experimentation to generate ML worthy AM data.

Materials Engineer: The Materials Engineer supports the Process Engineer in generating materials specific data of process, structure, or properties of interest. Their complimentary expertise lead to fulfilling ML requirements on AM data.

AM Scientist: The AM Scientist leads the research into ML-aided process development. Given that ML applications in AM are on the rise and their increased frequency has made it a hot research topic, familiarity with existing solutions and state of research is needed to optimize the path of development.

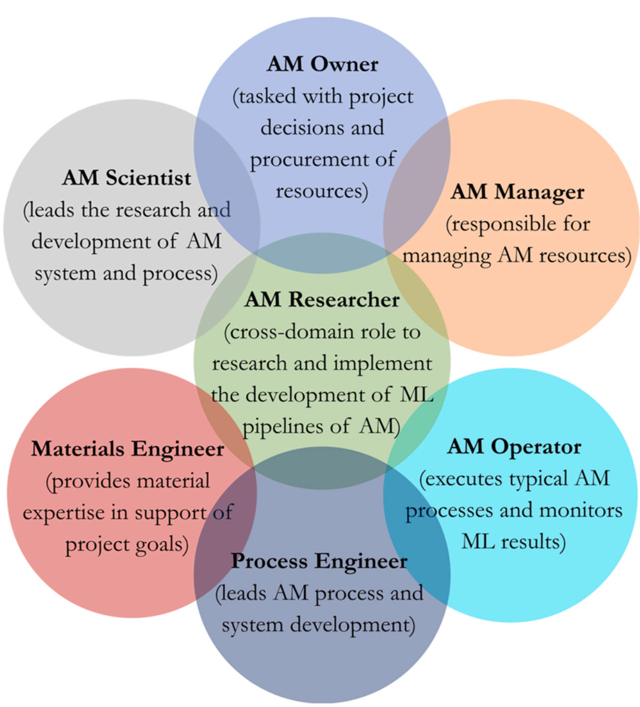

Fig. 4. AM roles with their descriptions and intersections.

Table 1
Required expertise of AM actors – AM0.

RQ#AM0	AM Actors	Requirements
A	AM Owner	Project management, leadership, business goals
B	AM Manager	Expertise in AM process and AM system technology
C	Process Engineer	Expertise in AM process technology
D	Materials Engineer	Expertise in materials science, and materials informatics for AM
E	AM Scientist	Expertise in ML methods and AM tasks
F	AM Researcher	Expertise in AM process technology and ML methods, familiarity with the state of data-driven AM research
G	AM Operator	Ability to use and monitor ML solutions for AM

AM Researcher: The AM Researcher is expected to have both AM and ML expertise to bridge the gap between two domains and use their cross-domain portfolio to guide AM and ML process development under the supervision of the AM Scientist.

AM Operator: The AM Operator is the end user of developed ML solutions of MLOps platform. AM Operator executes AM tasks and acts on ML results. The AM Operator also monitors the performance of ML solutions in the long run.

4.2. AM process development

While each AM technology requires specific steps to develop it for a given application, this section covers generic AM requirements to process development at industrial scale, which is applicable for all technologies to support ML development. Notably, the AM techniques with potential to rival subtractive manufacturing usually fall in the category of MAM, and the requirements discussed here cover major aspects of MAM processes. The process development step on the AM side is analogous to the ML development step, each involving certain research component to be built on top of the existing knowledge.

On the AM side, the critical requirement is to generate sufficient and representative AM data that can support the development of ML models. ML-aided research on AM has primarily been done at laboratory scale with little to no guidance regarding its applicability on a production machine. As a result, the process parameters chosen, and the ML tasks investigated often represent simplistic scenarios focused on validating ML applicability. This changes in an industrial setting where specific requirements need to be met and are evaluated against relatively stringent performance criteria (Eyers and Potter, 2017). The existing knowledge on AM technique is used as a reference to guide the exploration-exploitation phase that can lead to a suitable AM dataset. This existing knowledge needs to be complemented by further AM research. In this regard, the Process Engineer and the AM Researcher collaborate to come up with an experimental design that is representative of AM operations and is balanced with respect to the characteristics of interest. Currently, there is a lack of research on experimental designs that can cover the complexity of industrial operations. A compromise may be needed between available resources and the extent of experimentation possible. Once the process design is agreed upon, it is executed to generate in-situ, and ex-situ AM data. The data being generated needs to be resolved and captured correctly, so as to support ML development to an optimal extent. As a result, a portion of AM process development involves the selection and accurate installation of sensors that can capture the AM physical phenomena or characteristic of interest. The requirements to develop AM process are divided into process design, process models, and process iterations as highlighted in Table 2.

Table 2
AM process development tasks and sub tasks – AM1.

RQ#AM1	Process Development Tasks	Sub Tasks	Requirements
A	Process Design	Resources	Expertise in design of experiments for AM and man, material, & machine resources to carry out the experimental design (Tanco et al., 2007)
		Methods	Design of experiment models for AM (e.g., basic or advanced, simple or computational) (Astakhov, 2012)
B	Process Models	Models	Determine the appropriate model (e.g., analytical, numerical) type for the AM concern (e.g., defined by the relevant physical phenomenon) (Francois et al., 2017)
		Expertise	Expertise in domain-based AM modeling (e.g., mathematics, simulations) (Martukanitz et al., 2014)
C	Process Iterations	Materials	Build and/or substrate material to develop AM process (Seifi et al., 2016; Frazier, 2014)
		Systems	Sensor-integrated AM system (or validated computational models for synthetic data) to enable process development and data acquisition (Frazier, 2014; Gibson et al., 2021b)

4.3. AM system development

System refers to all configured standalone systems involved in the project, including the final printing setup used in the AM operations. For AM operations to meet the requirements of a working MLOps platform, the setup is integrated with sensors that capture the relevant physical phenomenon or characteristic of the AM process. Fig. 5 identifies AM hardware across all layers of the setup. In the hardware setup layer, we identify five basic systems needed to carry out the physical AM process, namely a material feeding system, a system for energy supply, a part, material or substrate handling/positioning system, an atmosphere control system and a cooling system for the whole setup. Their requirements are detailed in Table 3 and the functions are discussed as follows:

Material Feeding System: The material feeding system enables layer-wise supply of feedstock at the desired locations of deposition. Various material types and feeding setups exist depending on the AM technique used. The AM operations of material supply can be decomposed into material container, material transfer, and material deposition components at the system level.

System for Energy Supply: All AM techniques utilize energy directly (e.g., light, arc, laser, electron beam) or indirectly (e.g., power supply, binding force) or sometimes in both ways to accomplish the layer-wise build up leading to the finalized part. Energy delivery depends on the AM technique. For MAM, PBF and Directed Energy Deposition (DED) are two common methods.

Handling or Positioning System: The function of a handling system is similar to that of a typical subtractive manufacturing setup where feedstock is handled for a set of machining operations. In the case of AM, it usually concerns the execution of the scan strategy on the under-build component. The scan strategy depends on the application and can be roughly seen as a combination of tool path and deposition pattern.

Cooling System: The cooling system is inevitable for MAM techniques that involve high temperature operations. The cooling systems enables seamless supply of coolant to critical AM hardware (e.g., sensors and handing components) to maintain their normal operations.

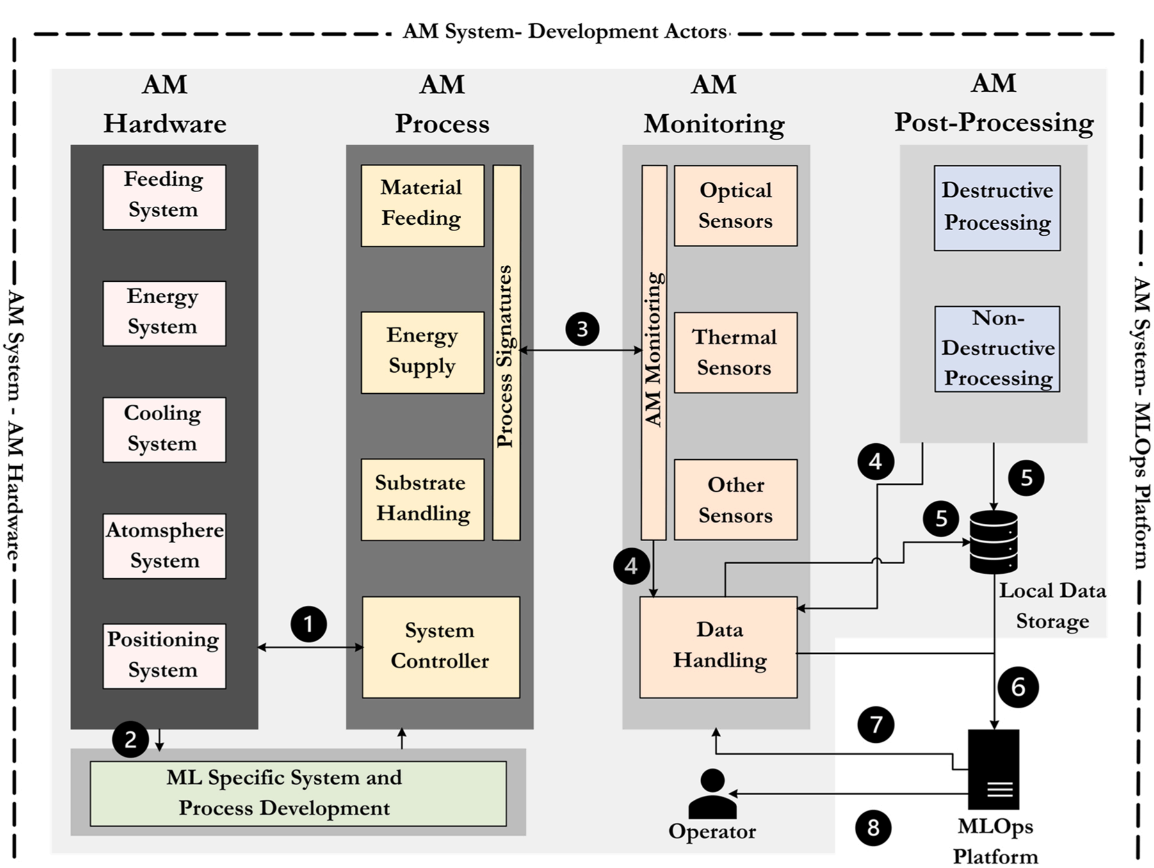

Fig. 5. Architectural components of AM system. Colors are used to identify systems belonging to a specific architectural layer. The numbered connections are explained in the text.

Table 3
Hardware components of a regular industrial MAM system and their requirements – AM2.

RQ#AM2	AM System	Requirements
A	Material Feeding System	The requirements for AM material feeding systems are dependent on the AM process technology. These requirements can be decomposed into material storage, material transfer and material deposition (National Academies of Sciences, 2019)
B	System for Energy Supply	Energy source and energy delivery unit are two required components of the AM energy systems, the specifications are closely related to AM process technology such as PBF or DED (Gibson et al. (2015a), (2015b)
C	Handling or Positioning System	AM systems are required to handle both incoming materials and under build part (e.g., part being printed and/or any substrate or base material) to support accurate layer-wise operations (Gibson et al. (2015a), (2015b)
D	Cooling System	A cooling source and a coolant distributor are two required components for colling different parts of the AM system (Gibson et al. (2015a), (2015b)
E	Atmosphere Control System	The requirements to control manufacturing atmosphere are usually dependent on the application for which an AM process technology is being used (Gibson et al., 2015a, 2015b), including preheating and interpass heating to control the whole manufacturing environment

Atmosphere Control System: The atmosphere control system is needed for applications (e.g., aerospace) where the AM setup requires shielding and enabling conditions (e.g., preheating) to accomplish manufacturing.

Fig. 5 identifies the inter- and intra-layer connections indicating exchange of data, information, or decisions among the systems. Connection 1 highlights the transfer of controls for an AM system and any information on their status in return. Connection 2 represents the knowledge usage for process and system development. Connection 3 covers the data acquisition from the process using sensors. Connection 4 relates to data handling operations on in-process and post-process data. Connection 5 connects the data handling model with local storage to register data. Connection 6 highlights the overall flow of data from storage to compute. The decisions on processed data are conveyed back to the AM system or the operator, depending on the nature of control, as depicted by Connection 7 and Connection 8, respectively.

In addition to hardware components required for normal AM operations, relevant sensors must be integrated with these components to capture key process signatures for data analytics and learning. We identify sensors separately from the above components and describe some representative types based on research trends in AM in support of ML applications. There are numerous ways to categorize all possible sensors that can be used to capture AM process phenomena. Table 4 highlights the sensor types and the associated generic requirements. Below are four most frequent types alongside their description which is adapted from the review of Zhu et al. (2021) on metal-based AM condition monitoring:

Sensors for Optical Signals: This category of sensors is used for the direct observation of the process phenomenon. These sensors are undoubtedly the most common types used to capture AM data. Several researchers have used images or videos of AM processes to investigate AM concerns. From ML applications perspective, the architectures of computer vision make the use of vision data popular in AM applications.

Sensors for Thermal Signals: This category of sensors is used to capture the thermal history or thermal characteristics of AM process phenomena. As indicated by numerous researchers, these signals can provide direct indication of the key process signatures (e.g., melt pool, layers) (Everton et al., 2016; Tapia and Elwany, 2014). Like optical signals, the resulting representations can be used for computer vision and geometric or 3D-based ML models (Cao et al., 2022).

Table 4
Generic requirements for AM process monitoring sensors – AM3.

RQ#AM3	Captured Signals	Captured Representation (s)	Common Sensor Types	General Sensing Requirements
A	Thermal Signals	Sequence, Graphic	Infrared Cameras, Pyrometer, Photodiode	Sensor setup (specified by the manufacturer), Calibration (dependent on the signal being captured), Sensor data handling (precise acquisition of data through resolution across the process) (Zhu et al., 2021; Lin et al., 2022)
B	Optical Signals	Graphic, 3D	CCD, CMOS, Imaging System, Inline Coherence, Laser Line Scanner	
C	Acoustic Signals	Sequence, Tabular, Graphics	Ultrasonic Transducer, Microphone	
D	Vibration Signals	Sequence, Tabular	Accelerometer	

Sensors for Acoustic Signals: This category of sensor is used to capture acoustic signals in a contact or non-contact fashion. Acoustic sensors could be used to capture certain AM process phenomenon (e.g., crack formation). The resulting data representation can have sequence, tabular, or graphic format implying the applicability of a range of ML models.

Sensors for Vibration Signals: Vibration sensors are another source of sequence data for capturing signals representative of unwanted process interruptions and system degradation.

In addition to 2D and 3D data captured by the previous sensor types, 1D signals from in-situ emissions can be captured. These 1D signals represent different physical spectra. The term sequence is used to imply a time ordering of captured data for these 1D signals. One exception to this arrangement of sensor types are videos coming from vision sensors which are also a sequence data type. There is a range of ML models that can capture temporal relations in AM sequence data making this data type of particular interest for long process dependencies ensuing from inherent layer-wise AM operations.

The IT components, sometimes referred to as the “computer” components (Gibson et al., 2015a), are an essential part of the AM system supporting informatics of AM process operations. Their role becomes more critical in the light of ML applications that are driven by data and information flow. In addition to the conventional AM controller that provides functionality to handle the substrate and execute programs, we group data and information-oriented functionalities in a data handling module supporting ML operations. The requirements for controller and handling module are detailed in Table 5. These modules are described below:

AM Controller Module: AM controller module, in its basic form, brings functionality in the AM system to execute a specific build. The AM Operator and Process Engineer can program a specific scan strategy and other parameters to fully automate the process of printing. Some research works in AM highlight a modification in the normal operations of an AM controller to support ML applications, such as registering data with reference to controller commands or information (Yang et al., 2021).

AM Data Handling Module: Data handling operations on the AM end represent the most critical requirements on IT systems to support MLOps. These operations can include spatial and temporal resolutions, data registering, pre-processing, and data structuring to prepare raw AM data for ML pipelines of feature engineering and model training.

Since AM is a unique domain for ML applications, some critical data is generated post-process once the build has been completed. This is a

Table 5
Requirements for AM process controller and AM data handling module – AM4.

RQ#AM4	IT System	IT Subsystem	Requirements
A	AM Process Controller	Program Generator (software on PC)	Ability to define process paths in workpiece coordinates and inverse kinematics module to calculate machine axis (Gibson et al. (2015a), (2015b)
		Program Consumer (machine or robot controller)	Program reading, path interpolation, kinematics calculations, and compensation (Gibson et al. (2015a), (2015b)
B	AM Data Handling	Acquisition Submodule	Application Programming Interface (APIs) to systematically collect data generated by different sensors (Liu et al., 2020a)
		Storage Submodule	Local or online storage capacity for AM process data (Liu et al., 2020a)
		Processing: Generic Submodule	Functionality to preprocess AM data, usually dependent on data type (e.g., graphic, tabular, sequence) (Zhang et al., 2022; Safdar et al., 2023a)
		Processing: AM-specific Submodule	Functionality to process AM data for specific ML application (e.g., fusion, alignment, registration) (Zhang et al., 2022; Safdar et al., 2023a)
		Transmission Submodule	Capacity to transmit AM data, information and decisions (within and across AM-ML interface)

fundamental difference from regular ML applications (e.g., computer vision) where labelled data (e.g., ImageNet (Deng et al., 2009)) is readily available, and the focus is given to algorithmic advancements. We identify these systems of AM data generation as coming from the well-known categories of destructive or non-destructive evaluation systems. Based on the trends and needs of AM, these post-process systems are divided into visual inspection, destructive, and non-destructive systems (Sreeraj et al., 2021) in Table 6 and are briefly described below:

Destructive Evaluation Systems: The destructive evaluation of printed parts is aimed at generating multi-scale information on material structure or properties. The generated data can be critical to supervision-based ML learning, often acting as labels or targets in ML terminology. The standard destructive testing systems belong to metallography procedures and mechanical testing.

Non-Destructive Evaluation Systems: Non-destructive evaluation procedures are also aimed at generating structure or property data of printed materials, but these systems can accomplish the task without the need to destroy or break the printed specimen. Common non-destructive systems in AM include X-Ray micro-Computed Tomography (CT), 3D scanning, Laser Ultrasonics and more as highlighted by group C of Table 6 of the AM requirements.

4.4. AM operations

The AM operations represent a set of practices from AM experts to carry out a particular build (Gibson et al., 2015b). These activities lead to data that is essential to ML training and development. We can divide the AM operations into physical and digital, depending on the activities involved. The physical operations represent standard operating procedures in AM, while the digital operations represent informatics of AM systems in support of AM and ML operations.

The physical AM operations are usually supervised by an Operator or

Table 6
Requirements for the destructive and non-destructive post-processing of AM. The generic requirements are listed against each evaluation method – AM5.

RQ#AM5	Post-Processing Techniques	Post-Processing Methodologies	Requirement
A	Visual Inspection	Post Process Inspection	Depending on the AM process technology and application, the requirements to visually inspect the process can include expertise, environment, and standard operating procedure to prepare the parts
B	Destructive Evaluation Systems	Metallography	Usually requires expertise in the process of metallography and the physical metallurgy of the material system being evaluated (Vander, 2012; Murr, 2018)
		Mechanical Testing	The requirement for mechanical testing depends on the type of test being done which is defined by the requisite of the application i.e., specification of the part, there are different types of mechanical tests, each usually requiring expertise on the technique and the testing equipment (Seifi et al., 2016)
C	Non-Destructive Evaluation Systems	Liquid Penetrant Testing	Require the liquid dye, the method is carried out in ultraviolet light with fluorescent dyes, it may also require surface preparation techniques and/or equipment (Sreeraj et al., 2021; Mandache, 2019)
		Ultrasonic Testing	Usually requires delicate equipment, the surface of the AM printed parts need to be prepared for inspection (Sreeraj et al., 2021; Mandache, 2019)
		Optical Methods	The requirements depend on the optical approach (e.g., laser scanning, structured light) used (Sreeraj et al., 2021; Mandache, 2019)
		Computed Tomography	Apart from the base CT system, its calibration can be one more requirement (Sreeraj et al., 2021; Mandache, 2019)
		Radiographic Techniques	Requirements depend on the radiographic medium used (e.g., Computed Radiography or Digital Radiography) (Sreeraj et al., 2021; Mandache, 2019)

Process Engineer. Since the hardware components don’t have to be set up every time, the physical operations usually start with validating the condition of the whole AM system. Once the system has been found to be in normal state, it is prepared for the operation. Depending on the AM technique, this step can be divided into several sub-steps. In metal AM, these steps can include part or material setup followed by the tool path planning and parameter selection. The process is then executed, and the process signatures of interest are monitored as part of the raw AM data generation. The second portion of AM data comes from the post-processing methods. The post-processing technique(s) depend on several factors, such as materials, applications, and characteristics of interest. The generated data is stored as historical process records to support the development of ML models. It is to be noted that while the AM operations remain largely same, post-processing needs may be significantly reduced (e.g., no need for the extensive experimental design for labeling) once the ML models are developed and deployed to produce real-time or near real-time predictions from AM data.

The AM informatics usually represent information flow for inter- and intra-connections of several components that are involved. We identify these as digital AM operations supporting AM and ML tasks. The input and output flow of AM systems usually represent these operations. The AM controller dissipates the set program and parameters during the AM process. This information can sometimes be integrated with downstream AM data to act as a reference for space or time coordinates during ML training. The enormous amount of data generated by the installed sensors needs to be handled. Research has been ongoing to develop methods and tools in support of AM process data. Similarly, solutions to efficiently post-process printed AM parts leading to AM data fall under the domain of materials informatics. The specific handling techniques employed depend on the in-situ and ex-situ representations being generated. Based on the needs of the project, the processed, resolved, and restructured data from the data handling module is stored on local drives or made available through cloud drives for ML process and system development. The final set of digital operations occurs between the AM and ML system once the MLOps platform has been deployed. On a generic scale, these operations represent outgoing flow of raw AM data and incoming flow of ML predictions. These interactions are described in detail in Section 6 on interface requirements. Since the MLOps platform is aimed at supporting AM process phase, the AM operations can be categorized into pre-processing, processing, and post-processing. The respective requirements are listed in the Table 7.

5. Requirements for machine learning

This section describes in detail the technical requirements for ML to support the MLOps for industrial applications of AM. These requirements are decomposed into actors, data & model pipeline development, system development, and critical ML operations. The requirements related to the interfaces are described separately in Section 6.

Table 7
Requirements on AM process operations – AM6.

RQ#AM6	Process Operations	Process Suboperations	Requirements
A	Pre-Processing	Part	Part level requirements include creating the digital design, converting it into AM ready format, setting up any substrate material (Gibson et al., 2015b)
		System	System level requirements include setup, calibration, validation check, and steps specific to the system model or manufacturer (Gibson et al., 2015b)
		Material	Preparing the AM material for uninterrupted and continuous supply during the manufacturing process, continuous material/feedstock validation (Gibson et al., 2015b)
		Plan	Selecting process parameters, deposition path and scan strategy (Gibson et al., 2015b)
B	Processing	Monitoring	Sensors, data acquisition modules, and enabling environment (Zhu et al., 2021; Lin et al., 2022)
		Control	Open- or closed-loop control modules (Zhu et al., 2021; Lin et al., 2022; Liu et al., 2020b)
C	Post-Processing	Evaluation	Visual, destructive or non-destructive evaluation systems (Sreeraj et al., 2021; Mandache, 2019)

5.1. ML actors

ML actors refer to the roles identified earlier. As can be seen, various experts must be involved in the development process. Fig. 6 identifies the key ML roles, describes their mandates, and highlights the interactions among these actors.

ML Owner: The ML Owner translates AM needs into actions for ML roles. The ML Owner is also responsible for conveying project level needs to the AM side. In addition to project level decisions, the ML Owner is responsible for making the necessary tools, materials, and resources available to all ML actors.

ML Architect: The ML Architect translates the technical requirement of handling AM data representation and supporting predictions on AM tasks into modules and engines with the relevant functionalities. The ML Architect also ensures that the ML operations can address the scale of AM needs. Depending on the existing state of AM system(s), the ML Architect is responsible for conveying the needs of ML.

Data Scientist: The primary role of the Data Scientist is to support the development of data engines that can contain relevant pipelines of processing and feature engineering. The Data Scientist also assists the ML Engineer in model development by preparing the data according to the requirements of ML models. Another important role of Data Scientist is to evaluate the AM data with respect to its quality and quantity and convey any concerns to the AM side during the development and the operation phases.

ML Engineer: The ML Engineer leads the development of ML architectures for AM tasks. The ML Engineer also connects ML roles in the light of AM needs owing to their cross-domain portfolio.

Development and Operations (DevOps) Engineer: The DevOps Engineer is responsible for implementing the architectural components of MLOps platform. Another key responsibility associated with the portfolio is to deploy the developed MLOps platform for training, educational, and operational support.

Software Engineer: The Software Engineer guides the DevOps Engineer during the development of MLOps platform. The Software Engineer provides guidelines to successfully scale ML operations for the platform being developed.

Table 8 summarises the required expertise of ML actors and groups them under ML0 as the first set of requirements.

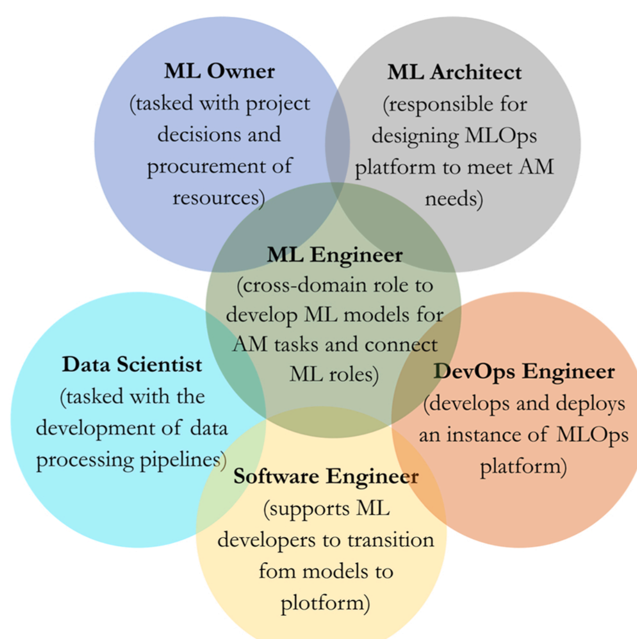

Fig. 6. ML roles with their descriptions and intersections.

Table 8
Required expertise of ML actors – ML0.

RQ#ML0	ML Actor	Requirements
A	ML Owner	Leadership, business goals, resource management
B	ML Architect	Expertise in ML project design and management
C	Data Scientist	Expertise in data science
D	ML Engineer	Expertise in ML development
E	DevOps Engineer	Expertise in support of MLOps
F	Software Engineer	Expertise in software engineering

5.2. ML process or pipeline development

The pipelines (pipeline on ML side is analogous to process on AM side) refer to integrated processing modules or units and can be divided into two categories namely data and model pipelines. Data pipelines are developed for specific AM data representations involved in the project whereas the requirements for models depend on the task at hand.

As per the trends in AM, the common AM data representations for ML applications were reviewed by Zhang et al. (2022) and tabular, graphic, 3D, and sequence-based representations were identified as the major types of AM data. Therefore, required data processing capabilities specific to these representations can be implemented as a pre-requisite to pipeline development. We propose to group these for each of the four categories of AM data namely tabular data engine, graphics data engine, 3D data engine, and sequence data engine. Each engine provides relevant libraries and feature engineering techniques to prepare data for ML applications. Table 9 indicates some of the major libraries for Python-based environments in this regard.

In addition to libraries, support for major data preparation techniques can be added in each engine. AM applications span diverse domains and data representations. The feature engineering techniques of AM can be grouped under five categories, namely subset selection, generation through transformation, generation through learning, knowledge-driven feature engineering, and integrated feature engineering (Safdar et al., 2023a). The frequency of use for each of the five technique varies from one data representation to another. In order to expedite the requirement on pipeline development, sample usage of these techniques can be added. Table 10 highlights the representative use of each technique against identified AM data representation.

Several of the reviews on ML applications in AM have highlighted the major model types. The models can be grouped based on different categories. While the recent literature highlights the trend of developing specific architecture that are difficult to generalise, ML models can be generally grouped into the categories of classification, regression, and clustering. Simpler models in each category can be made available for direct use. An additional category of hybrid models is added to account for scenarios where different models are integrated (e.g., ML-ML, ML-Numerical, ML-Optimal) for the same task. A comprehensive list of these base architectures and their potential applications is provided in Table 11. With the advances in data-driven AM research, more sophisticated architectures are being proposed nowadays. Moreover, the direct use of basic ML models can limit the freedom to dynamically update the model architectures and configuration, requiring integration with development tools and environments.

Though overlaps exist, the libraries for model development usually differ from the libraries for data processing. Table 12 highlights major libraries for shallow and deep ML models for Python-based environments. The choice of a specific library depends on several factors such as complexity of the approach, programming language, and project needs.

The development tools can include notebooks, integrated development environments (or IDEs) and version control systems. Python programming language provides a rich development ecosystem to support ML applications in AM. There exits several notebooks, IDEs, and version control systems that can support Python. Table 13 provides a brief list with the description of each tool.

Table 9
Required libraries to process AM data representations in support of AM tasks – ML1.

RQ#ML1	Data Representation	Library	Processing Function
A	Tabular	NumPy	To process AM data in matrices, vectors, and multi dimensional arrays from tabular representations (Harris et al., 2020)
		Matplotlib	To plot AM data (specifically tabular) in python supported frameworks (Hunter, 2007)
		Scipy	For scientific and technical computing of tabular representations from AM (Virtanen et al., 2020)
		Pandas	For handling AM numerical tables through provided operations and data structures (McKinney, 2010)
B	3D	Open3D	To process 3D data in diverse AM representations (e.g., point cloud, boundary representation, mesh) (Zhou et al., 2018)
		Binvox	Tool for binarizing 3D AM representation for ML model input
		Point Cloud Library	To process point cloud representations of AM (Rusu and Cousins, 2011)
		pyntcloud	To process 3D point cloud from AM design and process phases
		PyMeshLab	To process 3D triangular meshes of AM (Cignoni et al., 2008)
C	Graphics	OpenCV	Major library for handling graphic AM data (e.g., images, videos) (Bradski, 2000)
		Scikit-Image	To process AM images (Van der Walt et al., 2014)
		Pillow/PIL	To read and manipulate AM images (Umesh, 2012)
		Mahotas	To efficiently process AM images into NumPy arrays (Coelho, 2012)
		Pgmagick	For AM image modification and manipulation
		Albumentations	For fast image augmentations (Buslaev et al., 2020)
D	Sequence	Scipy Signal	To process digital AM signals in a linear or non linear fashion (Pajankar and Pajankar, 2017)
		ThinkDSP	AM digital signal processing in python (Downey, 2016)
		Librosa	To analyze AM sequence input from sensors (McFee et al., 2015)
		pyAudioAnalysis	To process and model AM audio signals (Giannakopoulos, 2015)

5.3. MLOps platform development

Platform refers to all the configured modules of the MLOps that can be packaged into one application. The architecture of the application can be divided into four layers namely interface, business or logic, development, and resource. A detailed description of these layers and their components are highlighted in Fig. 7.

Interface Layer: The interface layer is needed during the development and deployment of the system. This can be represented by a user interface and a data/decision interface. The user interface provides

Table 10
Required data preparation techniques for AM data representations – ML2.

RQ#ML2	Method\Data	Tabular	3D	Graphics	Sequence
A	Subset Selection Solorio-Fernández et al. (2020)	Selection of key parameters from the main set	Selection of relevant portions (e.g., voxels) from complete models	Selection of key regions (e.g., cropping) from the original images	Selection of significant subsequences from whole sequence
B	Generation through transformations Liu and Motoda (1998)	Mathematical transformation of tabular parameters	Simple transformations such as discretization and slicing	Conventional. Advanced, or Customized transformations	Time, frequency, or joint domain transformations
C	Generation through learning Goodfellow et al. (2016)	ML or DL based techniques (e.g., Autoencoders) to learn compact representation from original datasets			
D	Feature engineering with domain knowledge	The methods are subject to the context of domain knowledge being applied and are difficult to generalize			
E	Integrated feature engineering (Liu and Motoda, 1998)	Can integrate all four previous base techniques (e.g., ML2_A, ML2_B, ML3_C, ML4_D) for each data representation			

functionalities to support pipeline development operations such as access to development tools and resources. The user interface is also needed to communicate model decisions for open-loop systems.

AM Logic Layer: The AM Logic layer in Fig. 7 represents the business side as the sub-modules are tweaked to the needs of AM business logic. The modules can be grouped into two categories namely data modules and model modules. Data modules contain all relevant pipelines on AM data processing whereas the model modules contain all developed ML models for a specific AM business instance.

Development Layer: The development layer contains the functionalities discussed in the pipeline development section. These can be divided into feature engineering experiments and model development experiments. Feature engineering functionalities cover all of the data processing techniques whereas the model development functionalities provide libraries and development environment for ML models. A customized development environment, with support for programming languages and associated packages, can be made available to satisfy AM-based development needs of ML models.

Resource Layer: The resource layer of the system architecture groups both hardware and software resources needed to develop AM specific pipelines. Computation and storage represent the two important hardware resources for MLOps needs. A graphic processing unit or GPU of sufficient capacity (e.g., RAM) needs to be integrated with the MLOps. The storage is required for holding AM data and ML models. The software resources cover the data and model development tools discussed in the last section (Table 10 to Table 13).

The interface requirements for the MLOps platform can be categorized into three levels: within layers, across layers, and across platform boundaries. The interface within layers is used to access the functionalities of different submodules. The interface across the layers is used to carry data, information, or knowledge. The interface across the platform is used to connect with systems, actors, and resources. Table 14 lists the key requirements for the architectural layers of an MLOps platform.

5.4. ML operations

Operations refer to the practices to manage the lifecycle of ML models for AM tasks. These MLOps can be represented in terms of five key components such as development, deployment, monitoring, iteration, and governance. The key requirements of these MLOps in support of AM tasks are identified in this section. Table 15 lists these requirements for ML suboperations.

The development operations consist of activities that lead to ML solutions to AM challenges in the form of models or algorithms (John et al., 2021). These development operations require goals/tasks/metrics to be determined for the data processing and model development pipelines. Once the goals have been determined, the next phase of requirements concern methods and approaches that can be used to achieve the goals. The identified solutions are tuned to optimal in the development environment consisting of hardware and software components. Once the ML solutions to AM are developed, these are evaluated against the set goals. Before deploying the ML solution on AM printers, set practices on reproducibility and productionalization need to be followed for future development and re-use.

The deployment operation refers to making the developed ML solutions available for AM operations. The requirements to accomplishing this goal can be decomposed into hardware and software infrastructure. For industrial scale operations, automation in deployment is considered a key to avoiding errors and faults. In this regard, model availability and interface related functionalities can be embedded in ready-to-use tools and software (Garg et al., 2021). The capacity of compute and storage hardware depends on the nature of ML solutions (e.g., model and data size) and the way these solutions are used (e.g., closed-loop or open-loop).

The monitoring operations are based on several aspects grounded in concerns from different actors. The use of deployed resources needs to be monitored. This is done to ensure that the usage is not exceeding expected levels which could prompt data and model inspection. The data needs to be monitored for any changes in its distribution. In industrial environments, the data could change significantly from initial distribution prompting model updates (Bang et al., 2019; Xie et al., 2023). The ML Engineer and Data Scientist also monitor the model performance against the metrics of concern. A degrading performance could hint at any of the previous factors to be true which will require model update. The ML solutions are also monitored for their business value by the AM Manager and Owner to see if the expected return is obtained from the ML infrastructure.

The iterations are concerned with updates to existing solutions. A systematic and automated environment enabling both AM and ML actors to collaborate and generate new versions is required. The MLOps platform is interfaced with data sources and allows access to AM and ML actors for this purpose. The data processing and model training pipelines are reimplemented in accordance with the changes in data distributions, model performance, or business goals. A key requirement to accomplish this is through a feedback loop implemented in the system that can efficiently convey the concerns and subsequently the updates between AM and ML actors.

Governance operations are driven by ethical, legal and financial requirements. Out of these three, legal and financial requirements are more applicable to AM needs whereas the ethical requirements can be based on generic frameworks. The process and system development on both AM and ML sides can be used to represent these requirements. In this regard, we identify the data, models, and platform as three pillars driving governance concerns and initiatives. Data governance for ML based AM applications can follow the FAIR (findable, accessible, interoperable, and reusable) guiding principles (Wilkinson et al., 2016). Certain requirements governing data and models are domain guided and mutually agreed upon between AM and ML actors such as experimental design, dataset size, and model performance threshold. The platform development on both sides is governed by financial and operational constraints.

Table 11
Required ML models for major AM tasks. Some representative examples for each category are referenced – ML3.

RQ#ML3	Algorithm Category for AM Concerns	Algorithms	Requirements
A	Classification (e.g., categorical concerns such as porosity, cracks, defects, process domains, anomalies)	Decision Trees (Kingsford and Salzberg, 2008) (DT) Random Forest (Biau and Scornet, 2016) (RF) Feedforward Neural Networks (Svozil et al., 1997) (FFNN) Convolutional Neural Network (Albawi et al., 2017) (CNN) Support Vector Machines (Steinwart and Christmann, 2008) (SVM) K Nearest Neighbors (Kramer and Kramer, 2013) (kNN) Autoencoders (Pinaya et al., 2020) (AE) Recurrent Neural Networks (Medsker and Jain, 1999) (RNN) Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) (LSTM)	a) ML modeling expertise and proficiency in the theory of ML models b) Hardware and software resources to train, develop, deploy, and maintain ML models c) Relevant AM datasets from process design, process models, process monitoring and post-processing d) Machine learning and deep learning frameworks for efficient modeling
B	Regression (e.g., numeric concerns such as parameters, geometries, properties, characteristic values)	Linear Regression (Gross and Groß, 2003) (NR) Feedforward Neural Networks (Svozil et al., 1997) (FFNN) Gaussian Process Regression (Bishop and Nasrabadi, 2006) (GPR)	
C	Clustering (e.g., pattern or similarity-based concerns such as defects and anomalies)	K-means Clustering (Sinaga and Yang, 2020) (k-means) Self Organizing Maps (Kohonen, 2012) (SOM)	
D	Hybrid Approaches (e.g., complicated or data scarce AM concerns)	Hybrid ML (Guo et al., 2022) Mechanistic ML (Mozaffar et al., 2021) Optimal ML (Qin et al., 2020)	

6. Requirements for the interfaces

Given its multi-disciplinary nature, developing the MLOps platform for AM involves several interfaces. The interfaces can be categorized into human-human, human-system, and system-system interactions. The human-human interactions are guided by the project's needs to implement a specific version of the MLOps platform. The AM and ML roles discussed in the previous sections are analyzed here for bidirectional exchanges at a generic level. The human-system interactions occur in both AM and ML domains, where the respective actors of both sides take the lead in process and system development. The system-system interactions form the most sophisticated part of the interface and represent the informatics of the platform developed for an industrial AM system.

Table 12
ML and DL modeling framework for AM tasks supported by Python programming language – ML4.

RQ#ML4	Modeling Frameworks for AM	Description and Functionalities
A	PyTorch (Paszke et al., 2019)	A popular DL framework, provides data specific functionalities (Pytorch3D, TorchVision, TorchText, TorchAudio) and built-in debugging features among other features
B	Keras (Gulli and Pal, 2017)	Simple and easy to use DL framework for prototyping and experimentation
C	TensorFlow (Pang et al., 2020)	One of the most widely used frameworks for ML, provides functionality of both high- and low-level APIs.
D	Scikit-learn (Pedregosa et al., 2011)	Used for traditional machine learning tasks, provides functionality for tasks such as data processing, regression, classification and clustering
E	XGBoost	Specific for handling tabular and structure data, known for powerful gradient boosting framework
F	Caffe (Jia et al., 2014)	Focus on inference and training of computer vision-based DL models such as CNN
G	Theano (Al-Rfou et al., 2016)	Popular for efficient numerical computation of tabular data as multi dimensional arrays
H	Hugging Face (Wolf et al., 2019)	Popular for pre-trained transformer models, its vision transformer can assist in different AM tasks (e.g., segmentation)

Table 13
Required development tools for MLOps in support of AM tasks – ML5.

RQ#ML5	Development Tools for AM	Potential Tools	Requirements
A	Language (Raschka, 2015)	Python, R, Julia, Java	Relevant expertise in programming
B	Platform: Notebooks (Shen, 2014)	Jupyter Notebook, Google Collaboratory	Operating system, web browser, libraries and resources (compute and storage)
C	Platform: IDEs (Islam, 2015)	Visual Studio Code, Spyder, PyCharm, Rodeo	Operating system, libraries, and resources (compute and storage)
D	Code Management (Zolkifli et al., 2018)	GitHub, SVN, Mercurial	Relevant expertise in code management
E	Development Management/ Optimization	Optuna (Akiba et al., 2019), Hyperopt (Bergstra et al., 2013), Tune (Liaw et al., 2018)	Relevant expertise with specific libraries to optimize model development

6.1. Actor-actor (or human-human) interactions

The actor-actor interactions start right from the beginning and later go hand in hand to enable continuous development of the MLOps once the platform has been deployed. Fig. 8 represents all possible interactions of actors against specific roles identified earlier. The AM to ML interactions are primarily concerned with the as-is conditions of the AM system and requirements (or expectations) at the beginning. Later, the AM actors keep the ML side apprise of the state of deployed models or any bugs leading to incorrect predictions. Based on the identified AM requirements, the ML side conveys the hardware and software requirements needed for interface deployment on the industrial AM printer at the beginning. Once the instance of platform is populated with AM modules, the ML to AM interactions turn into validation,

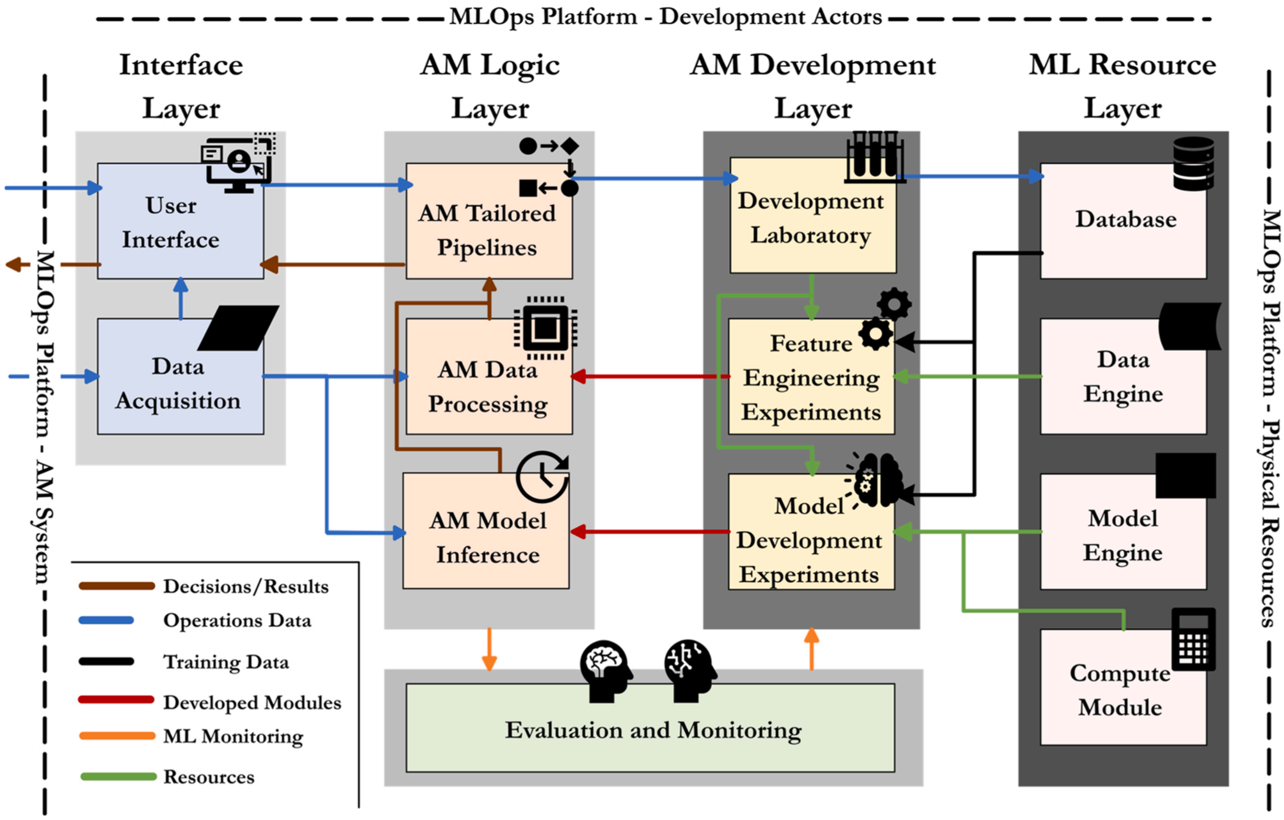

Fig. 7. Architectural components of MLOps system. Colors are used to identify systems belonging to a specific architectural layer.

Table 14
Requirements for architectural components to support MLOps platform – ML6.

RQ#ML6	Layer	Requirements for Layer Modules
A	Interface Kreuzberger et al. (2022)	*User:* Functionality to operate and/or use ML solutions *Application:* Capacity to handle AM data and information flow through APIs and submodules
B	Logic Kreuzberger et al. (2022)	*Pipelines:* Integration of data processing and ML models as per the AM requirements *Data Processing:* Tuned data processing functions for AM *Model Inference:* Trained ML models of AM
C	Development Kreuzberger et al. (2022)	*Environment:* Development Tools *Data Processing:* Data processing libraries and data processing methods *Model Development:* Modeling frameworks
D	Resource Kreuzberger et al. (2022)	*Storage-related:* Sufficient to store project data *Computation-related:* Sufficient to handle computation requirements of project data and models *Algorithms-related:* Support to develop ML models and process AM data

deployment, monitoring and update oriented discussions.

At the start of the project, the AMO conveys project level needs to the ML side, highlighted as AMN-O or *AM Needs by Owner*. These can include scale of manufacturing activities, general description of AM challenges, and expectations from ML solutions, among others. The AMM is able to convey detailed description of their as-is actions in support of AMO, dubbed as AMN-M (*AM needs by Manager*) in Fig. 8. This information is transformed into ML actions at the end of MLO (*ML Actions by Owner or MLA-O*) or MLA (*ML Actions by Architect or MLA-A*). These actions include planning and procuring generic ML resources to meet AM needs. Specific AM needs in the form of data, problem, and task are conveyed by the AM side through Process Engineer and Materials Engineer. AMN-PE or *AM Needs by Process Engineer* include description of AM operations and associated raw data in addition to expected support from the ML side. AMN-ME or *AM Needs by Materials Engineer* include materials data and informatics to describe the specific AM challenges. This key AM information is transformed into several ML actions. The description and state of AM data guides the DS to develop relevant functionalities into the system for subsequent development, highlighted as MLA-DS or *ML actions by Data Scientist*. These functionalities can include data processing engines for common AM representations such as images, videos, 3D scans, and sequences. The description of AM processes and operations support the MLE to develop modelling engine of the platform for catering intended ML tasks such as regression, classification, and clustering. AMR under the supervision of AMS conveys the research trends and state of existing solutions to drive MLA-E or *ML Actions by ML Engineer*. These three roles from both sides collaborate extensively to develop and implement ML solutions of AM.

Similarly, the ML to AM needs are conveyed starting with the *ML Needs by Owner* or MLN-O that encompass AM system updates, data generation requirements, and IT infrastructure to support the development of MLOps platform and its interfaces. MLN-O drives AMO and AMM to implement AM solutions in support of ML training and deployment. These include integration of sensors to AM systems, discussions on experiments for data generation and more. A specific version of these needs usually come from MLA, SE, and DevOpsE abbreviated as *ML Needs by Architect* (MLN-A), *ML Needs by Software Engineer* (MLN-SE), and *ML Needs by DevOps Engineer* (MLN-DE), respectively. Specifically,

Table 15
Requirements to support ML suboperations – ML7.

RQ#ML7	ML Operations	Requirements of Suboperations
A	Development Treveil et al. (2020)	*Goals*: Relevant AM expertise to set AM tasks *Methods*: Relevant modeling frameworks to develop ML models for the set tasks *Hardware*: Relevant development tools including compute and storage resources *Software:* Relevant development tools including languages, libraries, and environments *Reproducibility:* Relevant ML expertise to meet reproducibility requirements during development *Productionalization:* Relevant ML expertise to guide model development for downstream tasks
B	Deployment Treveil et al. (2020)	*Hardware:* Relevant deployment tools including compute and storage resources *Software:* Relevant deployment tools including languages, libraries, and environments
C	Monitoring Treveil et al. (2020)	*Resource:* Relevant AM and ML expertise for monitoring the usage of resources *Data:* Relevant AM and ML expertise for data distribution monitoring *Model:* Relevant AM and ML expertise for model performance monitoring
D	Iterations Treveil et al. (2020)	*Data:* Relevant AM process design and models for data update *Model:* Relevant ML modeling frameworks and development tools for model update
E	Governance Treveil et al. (2020)	*Data & Model:* Varying requirements depending on the ethical, legal, financial, and domain specific constraints in AM and ML

hardware requirements are conveyed, such as compatible sensor types, storage, and compute resources. The requirements from the last step drive AMO and AMM to arrange resources in support of ML development. MLE and DS jointly convey data specifications (e.g., quantity, quality) to AM side for developing the core of ML solutions in the form of specific architectures. *Needs by MLE and DS* (e.g., MLN-E and MLN-DS) drives AMR to support ME and PE for process and system development leading to valid experiments that can generate sufficient and representative data to support the ML development phase.

6.2. Actor-system (or human-system) interactions

The actor-system interactions take place on both AM and ML sides. On the AM side, these can be decomposed into the interactions during process development (PE-AMS or ME-AMS), system setup (PE-AMS), data generation (PE-AMS, AMR-AMS or ME-AMS), monitoring (Operator-AMS), and updates (the interactions depend on the nature of updates). For the ML side, actor-system interactions include MLOps platform development (DevOpsE-MLOps Platform), ML model development (MLE-MLOps Platform), feature engineering (DE-MLOps Platform), MLOps deployment (SE-MLOps Platform and DevOpsE-MLOps Platform), and updates (the interactions depend on the nature of updates). Since these interactions represent primary roles of actors from respective sides, their description remains the same as before, and can be found in Section 4.1 and Section 5.1.

6.3. System-system interactions

The system-system interactions cover the most critical component involved in the interface. While several interfaces can exist within the AM and ML systems, only the interactions between the two systems are considered in this section. Within the context of the interface, the fundamental requirement for the AM system is to seamlessly transmit ML ready data from the sensors and the AM system to the MLOps platform.

AM Data can be prepared in the data handling module of the AM system, where data acquisition, data registration, and data preprocessing can be implemented to process and restructure the incoming raw data. During the development process, this data is interfaced with the feature engineering pipelines directly or indirectly. In the former case, the data is saved or written directly in the storage system that ML development environment (e.g., Notebooks, IDEs) can access, whereas in the latter case the data is first stored offline in the standalone drives and then used in the development environments. Both versions of the pre-deployment interface are acceptable since no requirement exists on the frequency of incoming data streams. Once the developed modules are deployed, there is a need to consistently feed data for model inference (e.g., real-time or near-real-time). This is determined by the nature of actions performed on model results. Closed-loop systems have a stringent constraint on data availability to complete subsequent actions, whereas the open-loop systems produce the output for operators. In either case, the data can be first written in the drive of the deployed MLOps platform instance on edge device that is physically closer to the AM printer. This process is much faster than writing the data on regular cloud drives. The address of the incoming data being written is fed to the relevant ML model, accessed as API from the model repository of MLOps platform. The output of the model is either displayed on the operator's interface through a customized web interface (open-loop) or sent to the controller of the AM system to perform pre-programmed or rule-based actions (closed-loop).

The system-system interface needs to be made available through deployment. As a result, an instance of the developed MLOps platform is made available on an edge device. The edge computing serves to support faster inference for layer-to-layer predictions of the AM via developed models (Shi et al., 2016). Fig. 9 shows a potential arrangement of hardware to enable MLOps platform deployment near industrial AM printers. The available resources of CPUs, GPUs and storage drive(s) can be distributed among several virtual machines in support of training, maintenance, and update functionality. Table 16 lists the requirement to support deployment of the system-system interface.

7. Challenges and opportunities

Despite the detailed discussion on the requirements to support an MLOps platform, there are several challenges that need to be overcome to enable widespread adoption of ML in industry (Safdar et al., 2023b). The key challenges are identified below:

Knowledge Gap: Knowledge gap can be easily seen as the main challenge when developing such as cross-discipline platform. The requirements from different domains can be conflicting leading to confusion and misunderstanding. The frameworks and needs identified in this work can help researchers approach MLOps development in a systematic way, keeping in mind the pragmatic outcomes for the available resources and expertise.

Data Generation: Data generation is one fundamental requirement to get the system working on industrial AM printers. Big data is usually needed to train ML models. A systematic and comprehensive approach is needed to meet the data generation requirements so that both AM and ML constraints are fulfilled. Lack of data (e.g., quality, quantity, diversity) can hinder the platform development.

Resources: Resources are a major requirement to process and system development on AM and ML sides. However, procuring all the necessary

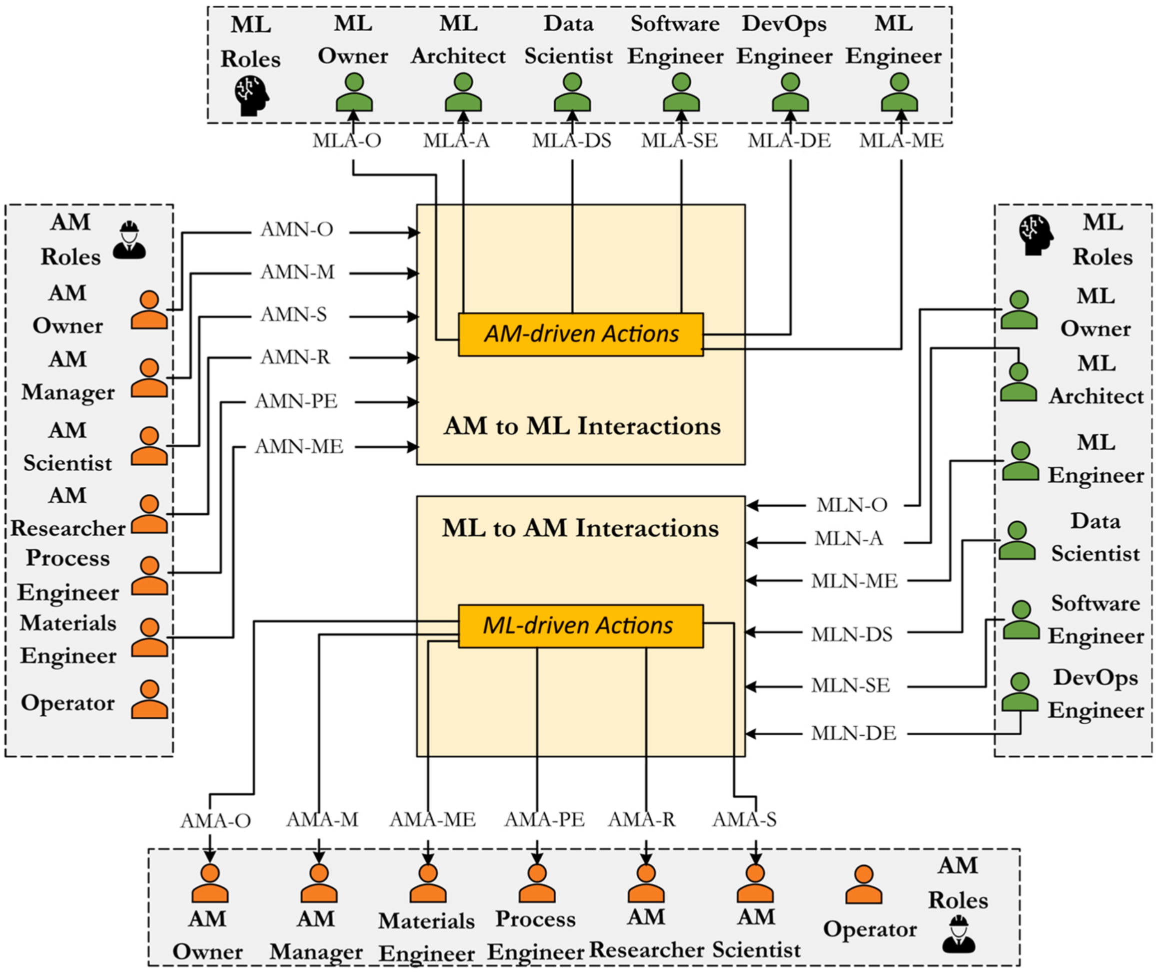

Fig. 8. Human-Human interactions indicating needs and actions for AM and ML actors in support of the MLOps platform.

resources can be difficult and challenging, especially for small enterprises. Additionally, expertise is needed to get diverse resources that are appropriate for the needs of ML and AM.

Reproducibility: Reproducibility is a common challenge for data-driven research. Cross-process, cross-system and cross-material application of ML solutions developed for the same tasks may require updates to data and model pipelines, the updates being often not straightforward.

The MLOps platform could enable defect free AM (Leberruyer et al., 2023). There are several opportunities brought by industrial adoption of data-driven AM which are listed below (Safdar et al., 2023b):

Quality Control: Quality control can be improved with the help of ML solutions. At the process phase, a range of AM concerns can be addressed. Anomalies can be identified, defects can be predicted, and properties and performance can be determined, all of which can be done in-situ with the support of deployed ML solutions.

Expedited Process Development: Expedited process development can be realized with empirical solutions of ML as opposed to parallel solutions requiring increased time and effort. The number of iterations needed to develop process-structure-property linkages are significantly reduced with the aid of data-driven tools. This also leads to a reduction in the lead time.

Cost Reduction: Cost reduction is one major motivation for pursing data-driven ML solutions. For instance, the material costs can be cut down by expediting the process development. Similarly, a close control on the quality leads to less parts being rejected in production.

Knowledge Creation and Transfer: Knowledge creation and transfer can be achieved with data-driven solutions. ML models of AM can be adapted to new applications and scenarios. These can also be updated when the same scenario is encountered with changes in original data distributions.

8. Conclusions and future works

The data driven solutions to AM are on the rise. There exist dozens of survey papers focused on summarizing these efforts. These models are scattered across the design, process, structure, and property phases. These lab scale efforts are rarely implemented and tested in industrial settings. As the solutions to AM problems mature, there is a need to deploy these in real-world operating conditions at industrial scale. However, there exists no guidance or framework to systematically adapt these solutions in industry. Moreover, the knowledge gap between AM and ML hinders smooth transition of these solutions from ML environments to AM machines. There is a lack of systematic approaches to guide this multi-disciplinary process involving several domains. Research into AM informatics is mostly limited to data and information management. With more and more data-driven models developed for AM challenges, a systems approach is needed to transition these solutions from academia and research to operations and industry.

We identified the key requirements for developing an MLOps platform that can be tailored to AM process needs and support ML models. A project level activity diagram has been presented and used to identify the requirements. The fundamental requirements were divided into roles, processes, systems, operations, and interfaces. The requirements were presented for a typical MAM process at industrial scale and are influenced by the trends in the data-driven AM research. The roles

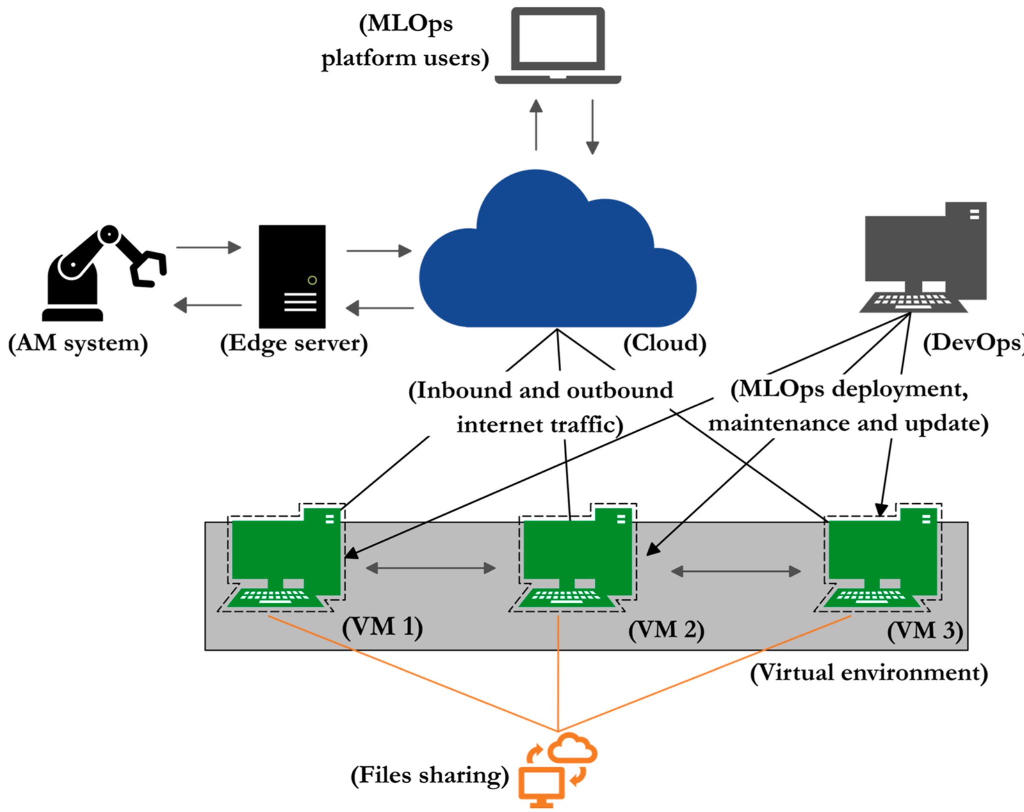

Fig. 9. Example scalable deployment of an MLOps platform (Braintoy's mlOS (BraintoyAI, 2023)) with multi-VM (virtual machine) architecture.

Table 16
Requirements to deploy a system-system interface or platform – ML8.

RQ#ML8	Components	Requirements
A	Infrastructure	Compute, storage, networking and security infrastructure for MLOps
B	Data Management	Functionality to acquire, store, process, and transfer AM data
C	Model Versioning	Functionality to update and manage ML models
D	Security	Adherence to compliance and security requirements (e.g., for data and model)
E	Monitoring and Update	Continuous monitoring and update capability
F	Integration	Seamless integration with AM (e.g., controller, database) and non-AM systems (e.g., users' interfaces)

represent the expertise needed on the ML and AM sides to meet the requirements of an MLOps platform. These also highlight the potential challenges of knowledge gaps and barriers that could be overcome with cross-domain expertise and close cooperation. The processes on AM side refer to domain specific models for AM process concerns. The processes on ML side refer to data and model development processes that are tailored to AM tasks and concerns. The systems on both sides differ greatly and the requirements for their development were detailed. For AM, these are primarily the base components and the added hardware in support of ML applications. For ML, the system is mainly composed of IT components comprising languages, libraries, models, interfaces, and resources. The operations on both sides follow established practices, consequently, we organized the requirements taking this aspect into account. Depending on the use case at hand, the entire pipeline of system setup, data generation, model development and deployment can be configured following the identified requirements.

In the future, the requirements of our proposed model can be expanded to include non-metallic AM technologies. Similarly, other AM lifecycle phases (e.g., design) can be considered. Eventually, relevant case studies can be implemented as proof of concept. There exits several MLOps platform that could be adapted to the needs of data-driven AM community (e.g., Braintoy's mlOS (BraintoyAI, 2023), DataRobot's MLOps (DataRobot, 2023), Amazon's SageMaker (Amazon, 2023)).

Funding

McGill Engineering Doctoral Award (MEDA) fellowship for Mutahar Safdar is acknowledged with gratitude. Mutahar Safdar also received financial support from National Research Council Canada (Grant# NRC INT-015-1).

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Mutahar Safdar reports financial support was provided by National Research Council Canada.

Data availability

No data was used for the research described in the article.

Acknowledgements

The authors are grateful to Digital Research Alliance of Canada (RRG# 4294) for providing computational resources to support this research. We would like to acknowledge the Artificial Intelligence Enhancement of Process Sensing for Adaptive Laser Additive Manufacturing (AI-SLAM) consortium between Canada and Germany for their support that made this work possible.

References

Akiba, T., et al. Optuna: A next-generation hyperparameter optimization framework. in Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019.
Al Mamun, A., et al., 2022. Securing cyber-physical additive manufacturing systems by in-situ process authentication using streamline video analysis. J. Manuf. Syst. 62, 429–440.
Al-Rfou, R.et al., Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints, 2016: p. arXiv: 1605.02688.
Albawi, S., Mohammed, T.A., Al-Zawi, S., 2017. Understanding of a convolutional neural network. 2017 international conference on engineering and technology (ICET). Ieee.
Amazon. Amazon SageMaker. 2023 [cited 2023 June 13]; Available from: ⟨https://aws.amazon.com/sagemaker/⟩.
Astakhov, V.P., 2012. Design of experiment methods in manufacturing: basics and practical applications. Stat. Comput. Tech. Manuf. 1–54.
ASTM, 2022. Additive manufacturing — General principles — Fundamentals and vocabulary. American Society of Testing Materials.
Bang, S.H., et al., 2019. A survey on knowledge transfer for manufacturing data analytics. Comput. Ind. 104, 116–130.
Baturynska, I., Martinsen, K., 2021. Prediction of geometry deviations in additive manufactured parts: comparison of linear regression with machine learning algorithms. J. Intell. Manuf. 32 (1), 179–200.
Bergstra, J., D. Yamins, and D.D. Cox. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. in Proceedings of the 12th Python in science conference. 2013. Citeseer.
Biau, G., Scornet, E., 2016. A random forest guided tour. Test 25, 197–227.
Bishop, C.M., Nasrabadi, N.M., 2006. Pattern recognition and machine learning, Vol. 4. Springer.
Bonnard, R., Hascoët, J.-Y., Mognol, P., 2019. Data model for additive manufacturing digital thread: state of the art and perspectives. Int. J. Comput. Integr. Manuf. 32 (12), 1170–1191.
Bradski, G., 2000. The openCV library. Dr. Dobb`S. J.: Softw. Tools Prof. Program. 25 (11), 120–123.
BraintoyAI. mlOS MLOps Platform. 2023 [cited 2023 Jun 13]; Available from: ⟨https://braintoy.ai/tour-the-platform/⟩.
Buslaev, A., et al., 2020. Albumentations: fast and flexible image augmentations. Information 11 (2), 125.
Cannizzaro, D., et al., *In-situ defect detection of metal Additive Manufacturing: an integrated framework*. IEEE Transactions on Emerging Topics in Computing, 2021.
Cao, W., et al., 2022. Geometric machine learning: research and applications. Multimed. Tools Appl. 81 (21), 30545–30597.
Charalampous, P., Kostavelis, I., Tzovaras, D., 2020. Non-destructive quality control methods in additive manufacturing: a survey. Rapid Prototyp. J.
Chen, X., et al., 2021. A review on wire-arc additive manufacturing: typical defects, detection approaches, and multisensor data fusion-based model. Int. J. Adv. Manuf. Technol. 117 (3), 707–727.
Cignoni, P., et al. *Meshlab: an open-source mesh processing tool.* in *Eurographics Italian chapter conference*. 2008. Salerno, Italy.
Coelho, L.P., *Mahotas: Open source software for scriptable computer vision.* arXiv preprint arXiv:1211.4907, 2012.
DataRobot. DataRobot AI Platform. 2023 [cited 2023 June 13]; Available from: ⟨https://www.datarobot.com/platform/⟩.
Deng, J., et al., 2009. Imagenet: A large-scale hierarchical image database. 2009 IEEE conference on computer vision and pattern recognition. Ieee.
Desai, P.S., Higgs III, C.F., 2019. Spreading process maps for powder-bed additive manufacturing derived from physics model-based machine learning. Metals 9 (11), 1176.
Dinar, M., Rosen, D.W., 2017. A design for additive manufacturing ontology. J. Comput. Inf. Sci. Eng. 17 (2).
Donegan, S.P., Schwalbach, E.J., Groeber, M.A., 2020. Zoning additive manufacturing process histories using unsupervised machine learning. Mater. Charact. 161, 110123.
Downey, A., 2016. Think DSP: digital signal processing in Python. O'Reilly Media, Inc.
Druzgalski, C., et al., 2020. Process optimization of complex geometries using feed forward control for laser powder bed fusion additive manufacturing. Addit. Manuf. 34, 101169.
Everton, S.K., et al., 2016. Review of in-situ process monitoring and in-situ metrology for metal additive manufacturing. Mater. Des. 95, 431–445.
Eyers, D.R., Potter, A.T., 2017. Industrial additive manufacturing: A manufacturing systems perspective. Comput. Ind. 92, 208–218.
Feng, S.C., et al., 2015. Fundamental requirements for data representations in laser-based powder bed fusion. in *International Manufacturing Science and Engineering Conference*. American Society of Mechanical Engineers.
Feng, S.C., et al., 2022. Additive manufacturing in situ and ex situ geometric data registration. J. Comput. Inf. Sci. Eng. 22 (6), 061003.
Feng, S.C., A.T. Jones, and Y. Lu. Microstructure and Mechanical Test Data Alignment for Additive Manufacturing Data Registration. in 2021 International Solid Freeform Fabrication Symposium. 2021. University of Texas at Austin.
Feng, S.C., Lu, Y., Jones, A.T., 2020. Measured data alignments for monitoring metal additive manufacturing processes using laser powder bed fusion methods. in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers.
Francois, M.M., et al., 2017. Modeling of additive manufacturing processes for metals: Challenges and opportunities. Curr. Opin. Solid State Mater. Sci. 21 (4), 198–206.
Frazier, W.E., 2014. Metal additive manufacturing: a review. J. Mater. Eng. Perform. 23 (6), 1917–1928.
Fu, Y., et al., 2022. Machine learning algorithms for defect detection in metal laser-based additive manufacturing: a review. J. Manuf. Process. 75, 693–710.
Gaikwad, A., et al., 2020. Heterogeneous sensing and scientific machine learning for quality assurance in laser powder bed fusion–A single-track study. Addit. Manuf. 36, 101659.
Garg, S., et al., 2021. On continuous integration/continuous delivery for automated deployment of machine learning models using mlops. 2021 IEEE fourth international conference on artificial intelligence and knowledge engineering (AIKE). IEEE.
Giannakopoulos, T., 2015. pyaudioanalysis: an open-source python library for audio signal analysis. PloS One 10 (12), e0144610.
Gibson, I., et al., 2021a. Additive manufacturing technologies, Vol. 17. Springer.
Gibson, I., et al., 2021b. Additive manufacturing technologies, Vol. 17. Springer.
Gibson, I., Rosen, D., Stucker, B., 2015a. In: Gibson, I., Rosen, D., Stucker, B. (Eds.), Powder Bed Fusion Processes, in Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing. Springer New York, New York, NY, pp. 107–145.
Gibson, I., Rosen, D., Stucker, B., 2015b. In: Gibson, I., Rosen, D., Stucker, B. (Eds.), Directed Energy Deposition Processes, in Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing. Springer New York, New York, NY, pp. 245–268.
Gibson, I., Rosen, D., Stucker, B., 2015a. In: Gibson, I., Rosen, D., Stucker, B. (Eds.), Development of Additive Manufacturing Technology, in Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing. Springer New York, New York, NY, pp. 19–42.
Gibson, I., Rosen, D., Stucker, B., 2015a. In: Gibson, I., Rosen, D., Stucker, B. (Eds.), Software Issues for Additive Manufacturing, in Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing. Springer New York, New York, NY, pp. 351–374.
Gibson, I., Rosen, D., Stucker, B., 2015b. In: Gibson, I., Rosen, D., Stucker, B. (Eds.), Introduction and Basic Principles, in Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing. Springer New York, New York, NY, pp. 1–18.
Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning. MIT press.
Grasso, M., Gallina, F., Colosimo, B.M., 2018. Data fusion methods for statistical process monitoring and quality characterization in metal additive manufacturing. Procedia CIRP 75, 103–107.
Gräßler, I., et al. Data Management for additive manufacturing: survey on requirements and current state. in DS 84: Proceedings of the DESIGN 2016 14th International Design Conference. 2016.
Gribova, V., et al. The concept of support for laser-based additive manufacturing on the basis of artificial intelligence methods. in Artificial Intelligence: 18th Russian Conference, RCAI 2020, Moscow, Russia, October 10–16, 2020, Proceedings 18. 2020. Springer.
Gross, J., Groß, J., 2003. Linear regression, Vol. 175. Springer Science & Business Media.
Gulli, A., Pal, S., 2017. Deep learning with Keras. Packt Publishing Ltd.
Guo, S., et al., 2022. Machine learning for metal additive manufacturing: towards a physics-informed data-driven paradigm. J. Manuf. Syst. 62, 145–163.
Harris, C.R., et al., 2020. Array programming with NumPy. Nature 585 (7825), 357–362.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.
Hu, C., et al., 2021. The fabrication of long carbon fiber reinforced polylactic acid composites via fused deposition modelling: experimental analysis and machine learning. J. Compos. Mater. 55 (11), 1459–1472.
Hunter, J.D., 2007. Matplotlib: a 2D graphics environment. Comput. Sci. Eng. 9 (03), 90–95.
Islam, Q.N., 2015. Mastering PyCharm. Packt Publishing Ltd.
Jia, Y., et al., 2014. Caffe: convolutional architecture for fast feature embedding. Proc. 22nd ACM Int. Conf. Multimed.
Johnson, N., et al., 2020. Invited review: machine learning for materials developments in metals additive manufacturing. Addit. Manuf. 36, 101641.
John, M.M., Olsson, H.H., Bosch, J., 2021. Towards mlops: A framework and maturity model. 2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE.
Joshi, M., et al. *Applications of Supervised Machine Learning Algorithms in Additive Manufacturing: A Review*. in 2019 International Solid Freeform Fabrication Symposium. 2019. University of Texas at Austin.
Khairallah, S.A., et al., 2020. Controlling interdependent meso-nanosecond dynamics and defect generation in metal 3D printing. Science 368 (6491), 660–665.
Kim, D.B., et al., 2015. Streamlining the additive manufacturing digital spectrum: a systems approach. Addit. Manuf. 5, 20–30.
Kim, D.B., et al., 2017. Toward a digital thread and data package for metals-additive manufacturing. Smart Sustain. Manuf. Syst. 1 (1), 75.
Kim, D.B., 2019. An approach for composing predictive models from disparate knowledge sources in smart manufacturing environments. J. Intell. Manuf. 30 (4), 1999–2012.
Kingsford, C., Salzberg, S.L., 2008. What are decision trees? Nat. Biotechnol. 26 (9), 1011–1013.
Ko, H., et al., 2021. Machine learning and knowledge graph based design rule construction for additive manufacturing. Addit. Manuf. 37, 101620.
Ko, H., et al., 2019. Machine learning based continuous knowledge engineering for additive manufacturing. 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE). IEEE.
Kohonen, T., 2012. Self-organizing maps, Vol. 30. Springer Science & Business Media.
Kramer, O., Kramer, O., 2013. K-nearest neighbors. Dimens. Reduct. unsupervised nearest neighbors 13–23.

Kreuzberger, D., Kühl, N., Hirschl, S., 2022. Machine learning operations (mlops): Overview, definition, and architecture. arXiv Prepr. arXiv:2205. 02302.
Kumar, H.A., et al., 2021. Faster temperature prediction in the powder bed fusion process through the development of a surrogate model. Opt. Laser Technol. 141, 107122.
Leberruyer, N., et al., 2023. Toward zero defect manufacturing with the support of artificial intelligence—insights from an industrial application. Comput. Ind. 147, 103877.
Lee, J., et al., 2021. Review on quality control methods in metal additive manufacturing. Appl. Sci. 11 (4), 1966.
Lhachemi, H., Malik, A., Shorten, R., 2019. augmented reality, cyber-physical systems, and feedback control for additive manufacturing: a review. IEEE Access 7, 50119–50135.
Liang, J.S., 2018. An ontology-oriented knowledge methodology for process planning in additive layer manufacturing. Robot. Comput. -Integr. Manuf. 53, 28–44.
Liaw, R., et al., Tune: A research platform for distributed model selection and training. arXiv preprint arXiv:1807.05118, 2018.
Lin, X., et al., 2022. Metal-based additive manufacturing condition monitoring methods: from measurement to control. ISA Trans. 120, 147–166.
Liu, G., et al., 2021a. Additive manufacturing of structural materials. Mater. Sci. Eng.: R: Rep. 145, 100596.
Liu, C., et al., 2020a. Digital twin-enabled collaborative data management for metal additive manufacturing systems. J. Manuf. Syst.
Liu, C., et al., 2020b. Machine Learning-enabled feedback loops for metal powder bed fusion additive manufacturing. Procedia Comput. Sci. 176, 2586–2595.
Liu, S., et al., 2021b. Machine learning for knowledge transfer across multiple metals additive manufacturing printers. Addit. Manuf. 39, 101877.
Liu, H., Motoda, H., 1998. Feature extraction, construction and selection: A data mining perspective, Vol. 453. Springer Science & Business Media.
Lu, Y., Choi, S., Witherell, P., 2015. Towards an integrated data schema design for additive manufacturing: conceptual modeling. in *International Design Engineering Technical Conferences and Computers and Information In Engineering Conference*. American Society of Mechanical Engineers,.
Lu, Y., Witherell, P., Donmez, A., 2017. A collaborative data management system for additive manufacturing. in International design engineering technical conferences and computers and information in engineering conference. American Society of Mechanical Engineers.
Lu, Y., et al., 2020. Camera-based coaxial melt pool monitoring data registration for laser powder bed fusion additive manufacturing. in *ASME International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers.
Mahmoud, D., et al., 2021. Applications of machine learning in process monitoring and controls of L-PBF additive manufacturing: a review. Appl. Sci. 11 (24), 11910.
Mandache, C., 2019. Overview of non-destructive evaluation techniques for metal-based additive manufacturing. Mater. Sci. Technol. 35 (9), 1007–1015.
Martukanitz, R., et al., 2014. Toward an integrated computational system for describing the additive manufacturing process for metallic materials. Addit. Manuf. 1, 52–63.
Mativo, T., Fritz, C., Fidan, I., 2018. Cyber acoustic analysis of additively manufactured objects. Int. J. Adv. Manuf. Technol. 96 (1), 581–586.
McFee, B., et al., 2015. librosa: audio and music signal analysis in python. Proc. 14th python Sci. Conf.
McKinney, W. Data structures for statistical computing in python. in Proceedings of the 9th Python in Science Conference. 2010. Austin, TX.
Medsker, L., Jain, L.C., 1999. Recurrent neural networks: design and applications. CRC press.
Mehrpouya, M., et al., 2019. The potential of additive manufacturing in the smart factory industrial 4.0: a review. Appl. Sci. 9 (18), 3865.
Meng, L., et al., 2020. Machine learning in additive manufacturing: a review. Jom 72 (6), 2363–2377.
Mies, D., Marsden, W., Warde, S., 2016. Overview of additive manufacturing informatics: "a digital thread". Integr. Mater. Manuf. Innov. 5 (1), 114–142.
Moltumyr, A.H., Arbo, M.H., Gravdahl, J.T., 2020. Towards vision-based closed-loop additive manufacturing: a review. 2020 3rd International Symposium on Small-scale Intelligent Manufacturing Systems (SIMS). IEEE.
Mozaffar, M., et al., 2021. Mechanistic artificial intelligence (mechanistic-AI) for modeling, design, and control of advanced manufacturing processes: Current state and perspectives. J. Mater. Process. Technol., 117485
Murr, L.E., 2018. A metallographic review of 3D printing/additive manufacturing of metal and alloy products and components. Metallogr., Microstruct., Anal. 7, 103–132.
Muthiah, C., et al., 2022. Application of machine learning in fused deposition modeling: A review. in AIP Conference Proceedings. AIP Publishing LLC,.
National Academies of Sciences, E. and Medicine, Data-Driven Modeling for Additive Manufacturing of Metals: Proceedings of a Workshop. 2019: National Academies Press.
Nguyen, L., Buhl, J., Bambach, M., 2020. Continuous Eulerian tool path strategies for wire-arc additive manufacturing of rib-web structures with machine-learning-based adaptive void filling. Addit. Manuf. 35, 101265.
Ogoke, F., Farimani, A.B., 2021. Thermal control of laser powder bed fusion using deep reinforcement learning. Addit. Manuf. 46, 102033.
Oleff, A., et al., 2021. Process monitoring for material extrusion additive manufacturing: a state-of-the-art review. Prog. Addit. Manuf. 1–26.
Pajankar, A., Pajankar, A., 2017. Signal processing with SciPy. Raspberry Pi Supercomput. Sci. Program.: MPI4PY, NumPy, SciPy Enthus. 139–147.
Pang, B., Nijkamp, E., Wu, Y.N., 2020. Deep learning with tensorflow: a review. J. Educ. Behav. Stat. 45 (2), 227–248.
Park, H., et al., 2019. A framework for identifying and prioritizing data analytics opportunities in additive manufacturing. 2019 IEEE international conference on big data (Big Data). IEEE,.
Paszke, A., et al., 2019. Pytorch: an imperative style, high-performance deep learning library. Adv. Neural Inf. Process. Syst. 32.
Pedregosa, F., et al., 2011. Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.
Phua, A., Davies, C., Delaney, G., 2022. A digital twin hierarchy for metal additive manufacturing. Comput. Ind. 140, 103667.
Piller, F.T., Weller, C., Kleer, R., 2015. Business models with additive manufacturing—opportunities and challenges from the perspective of economics and management. Springer International Publishing.
Pinaya, W.H.L., et al., 2020. Autoencoders. in Machine learning. Elsevier, pp. 193–208.
Qin, J., et al., 2022. Research and application of machine learning for additive manufacturing. Addit. Manuf., 102691
Qi, X., et al., 2019. Applying neural-network-based machine learning to additive manufacturing: current applications, challenges, and future perspectives. Engineering 5 (4), 721–729.
Qin, Y., et al., 2019. Status, comparison, and future of the representations of additive manufacturing data. Comput. -Aided Des. 111, 44–64.
Qin, J., et al., 2020. Deep learning-driven particle swarm optimisation for additive manufacturing energy optimisation. J. Clean. Prod. 245, 118702.
Raschka, S., 2015. Python machine learning. Packt publishing ltd.
Renken, V., et al., 2019. In-process closed-loop control for stabilising the melt pool temperature in selective laser melting. Prog. Addit. Manuf. 4 (4), 411–421.
Roh, B.-M., et al., 2021. Ontology-based process map for metal additive manufacturing. J. Mater. Eng. Perform. 30 (12), 8784–8797.
Roh, B.-M., et al., 2016. Ontology-based laser and thermal metamodels for metal-based additive manufacturing. in International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers.
Roy, M., Wodo, O., 2020. Data-driven modeling of thermal history in additive manufacturing. Addit. Manuf. 32, 101017.
Rusu, R.B., Cousins, S., 2011. 3d is here: Point cloud library (pcl). 2011 IEEE international conference on robotics and automation. IEEE.
Răileanu, S., et al., 2018. A cloud-based manufacturing control system with data integration from multiple autonomous agents. Comput. Ind. 102, 50–61.
Safdar, M., et al., 2023a. In: Safdar, M., et al. (Eds.), Introduction, in Engineering of Additive Manufacturing Features for Data-Driven Solutions: Sources, Techniques, Pipelines, and Applications. Springer Nature Switzerland, Cham, pp. 1–15.
Safdar, M., et al., 2023a. Feature Engineering in Additive Manufacturing. In: Safdar, M., et al. (Eds.), Engineering of Additive Manufacturing Features for Data-Driven Solutions: Sources, Techniques, Pipelines, and Applications. Springer Nature Switzerland, Cham, pp. 17–43.
Safdar, M., et al., 2023a. Engineering of Additive Manufacturing Features for Data-Driven Solutions: Sources, Techniques, Pipelines, and Applications. Springer Nature.
Safdar, M., et al., 2023b. Challenges and opportunities in additive manufacturing data preparation. In: Safdar, M., et al. (Eds.), in *Engineering of Additive Manufacturing Features for Data-Driven Solutions: Sources, Techniques, Pipelines, and Applications*. Springer Nature Switzerland, Cham, pp. 135–138.
Sanfilippo, E.M., Belkadi, F., Bernard, A., 2019. Ontology-based knowledge representation for additive manufacturing. Comput. Ind. 109, 182–194.
Scime, L., Beuth, J., 2018. Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm. Addit. Manuf. 19, 114–126.
Seifi, M., et al., 2016. Overview of materials qualification needs for metal additive manufacturing. Jom 68, 747–764.
Shen, H., 2014. Interactive notebooks: sharing the code. Nature 515 (7525), 152.
Shi, R., et al., 2020. Microstructural control in metal laser powder bed fusion additive manufacturing using laser beam shaping strategy. Acta Mater. 184, 284–305.
Shi, W., et al., 2016. Edge computing: vision and challenges. IEEE Internet things J. 3 (5), 637–646.
Sinaga, K.P., Yang, M.-S., 2020. Unsupervised K-means clustering algorithm. IEEE Access 8, 80716–80727.
Solorio-Fernández, S., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., 2020. A review of unsupervised feature selection methods. Artif. Intell. Rev. 53 (2), 907–948.
Sreeraj, P., Mishra, S.K., Singh, P.K., 2021. A review on non-destructive evaluation and characterization of additively manufactured components. Prog. Addit. Manuf. 1–24.
Steinwart, I., Christmann, A., 2008. Support vector machines. Springer Science & Business Media.
Svozil, D., Kvasnicka, V., Pospichal, J., 1997. Introduction to multi-layer feed-forward neural networks. Chemom. Intell. Lab. Syst. 39 (1), 43–62.
Tanco, M., et al. *Manufacturing industries need Design of Experiments (DoE)*. in *World Congress on Engineering*. 2007.
Tapia, G., Elwany, A., 2014. A review on process monitoring and control in metal-based additive manufacturing. J. Manuf. Sci. Eng. 136 (6).
Thomas, M., et al., 2020. Distributed heterogeneous compute infrastructure for the study of additive manufacturing systems. MRS Adv. 5 (29), 1547–1555.
Tian, C., et al., 2021. Data-driven approaches toward smarter additive manufacturing. Adv. Intell. Syst. 3 (12), 2100014.
Treveil, M., et al., 2020. Introducing MLOps. O'Reilly Media.
Umesh, P., 2012. Image processing in python. CSI Commun. 23, 2.
Valizadeh, M., Wolff, S.J., 2022. Convolutional neural network applications in additive manufacturing: a review. Adv. Ind. Manuf. Eng., 100072
Vander, G.F., 2012. Applied metallography. Springer Science & Business Media.

Virtanen, P., et al., 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods 17 (3), 261–272.
Van der Walt, S., et al., 2014. scikit-image: image processing in Python. PeerJ 2, e453.
Wang, Z., et al., 2022. Data-driven modeling of process, structure and property in additive manufacturing: a review and future directions. J. Manuf. Process. 77, 13–31.
Wang, Y., et al., 2019. Production planning for cloud-based additive manufacturing—A computer vision-based approach. Robot. Comput. -Integr. Manuf. 58, 145–157.
Wang, Q., et al., 2020. Model-based feedforward control of laser powder bed fusion additive manufacturing. Addit. Manuf. 31, 100985.
Wilkinson, M.D., et al., 2016. The FAIR guiding principles for scientific data management and stewardship. Sci. data 3 (1), 1–9.
Wolf, T., et al., 2019. *Huggingface's. Transform.: State—Art. Nat. Lang. Process.* arXiv Prepr. arXiv 1910, 03771.
Wu, M., Song, Z., Moon, Y.B., 2019b. Detecting cyber-physical attacks in CyberManufacturing systems with machine learning methods. J. Intell. Manuf. 30 (3), 1111–1123.
Wu, H., Yu, Z., Wang, Y., 2019a. Experimental study of the process failure diagnosis in additive manufacturing based on acoustic emission. Measurement 136, 445–453.
Wu, B., et al., 2021. In situ monitoring methods for selective laser melting additive manufacturing process based on images—A review. China Foundry 18 (4), 265–285.
Xie, J., et al., 2023. A sequential cross-product knowledge accumulation, extraction and transfer framework for machine learning-based production process modelling. Int. J. Prod. Res. 1–21.
Xiong, Y., et al., 2020. A knowledge-based process planning framework for wire arc additive manufacturing. Adv. Eng. Inform. 45, 101135.
Yang, Z., et al., 2021. In-process data fusion for process monitoring and control of metal additive manufacturing. in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers,.
Zhang, Y., et al., 2022. A systematic review on data of additive manufacturing for machine learning applications: the data quality, type, preprocessing, and management. J. Intell. Manuf. 1–36.
Zhang, Y., et al., 2021a. Predictive manufacturability assessment system for laser powder bed fusion based on a hybrid machine learning model. Addit. Manuf. 41, 101946.
Zhang, Y., Yan, W., 2022. Applications of machine learning in metal powder-bed fusion in-process monitoring and control: status and challenges. J. Intell. Manuf. 1–24.
Zhang, H., et al., 2021b. A knowledge transfer framework to support rapid process modeling in aerosol jet printing. Adv. Eng. Inform. 48, 101264.
Zhou, Q.-Y., J. Park, and V. Koltun, *Open3D: A modern library for 3D data processing*. arXiv preprint arXiv:1801.09847, 2018.
Zhu, K., Fuh, J.Y.H., Lin, X., 2021. Metal-based additive manufacturing condition monitoring: a review on machine learning based approaches. IEEE/ASME Trans. Mechatron.
Zohdi, T., 2019. Electrodynamic machine-learning-enhanced fault-tolerance of robotic free-form printing of complex mixtures. Comput. Mech. 63, 913–929.
Zolkifli, N.N., Ngah, A., Deraman, A., 2018. Version control system: a review. Procedia Comput. Sci. 135, 408–415.