Security Risks and Best Practices of MLOps: A Multivocal Literature Review

Fabio Calefato[1], Filippo Lanubile[1] and Luigi Quaranta[1]

[1]*University of Bari, Bari, Italy*

Abstract

MLOps practices and tools are designed to streamline the deployment and maintenance of production-grade ML-enabled systems. As with any software workflow and component, they are susceptible to various security threats. In this paper, we present a Multivocal Literature Review (MLR) aimed at gauging current knowledge of the risks associated with the implementation of MLOps processes and the best practices recommended for their mitigation. By analyzing a varied range of sources of academic papers and non-peer-reviewed technical articles, we synthesize 15 risks and 27 related best practices, which we categorized into 8 themes. We find that while some of the risks are known security threats that can be mitigated through well-established cybersecurity best practices, others represent MLOps-specific risks, mostly concerning the management of data and models.

Keywords

MLSecOps, SecMLOps, machine learning, deep learning, cybersecurity

1. Introduction

During the last few years, the pervasive adoption of data-driven Artificial Intelligence (AI) across a wide range of application domains has led to the proliferation of commercial Machine Learning (ML)-enabled systems. Consequently, there has been a push to develop mature practices and tools for the deployment and maintenance of ML components in production. This collective effort has given rise to a burgeoning research and practice field known as MLOps (Machine Learning Operations). Rooted in Software Engineering and inspired by DevOps [1], MLOps places emphasis on process automation to achieve continuous delivery of ML models within ML-enabled systems [2]. Among the several objectives of MLOps, there is the facilitation of a number of non-functional requirements of ML-enabled systems (e.g., reproducibility, explainability, fairness), among which system security is considered of growing importance, especially when ML is deployed in safety- or mission-critical domains.

The overall security of ML-enabled systems depends on several intricate factors, and MLOps workflows themselves can be the source of dangerous security holes. In this study, we aim to assess the current level of understanding of MLOps security from the perspective of researchers and practitioners. Consequently, we define the following research questions:

ITASEC 2024: The Italian Conference on CyberSecurity, April 08–12, 2024, Salerno, Italy

fabio.calefato@uniba.it (F. Calefato); filippo.lanubile@uniba.it (F. Lanubile); luigi.quaranta@uniba.it (L. Quaranta)

https://collab.di.uniba.it/fabio/ (F. Calefato); https://collab.di.uniba.it/lanubile/ (F. Lanubile); https://collab.di.uniba.it/luigi-quaranta/ (L. Quaranta)

0000-0003-2654-1588 (F. Calefato); 0000-0003-3373-7589 (F. Lanubile); 0000-0002-9221-0739 (L. Quaranta)

RQ1 Is there a widely accepted definition of MLOps security in the literature?

RQ2 What are the risks that affect ML-enabled systems and their development pipelines?

RQ3 What best practices can MLOps practitioners implement to mitigate security risks?

To answer these questions, we conducted a Multivocal Literature Review (MLR), i.e., a form of Systematic Literature Review (SLR) in which gray literature – i.e., non-peer-reviewed sources, like technical blog posts – are analyzed alongside white literature – i.e., academic papers – to thoroughly explore a novel (and potentially under-researched) topic of interest.

By analyzing selected articles, we identified 15 risks and 27 best practices related to MLOps security, which we classified into 8 themes. Overall, the small number of sources recovered suggests that general knowledge and awareness of MLOps security may still be in its early stages of development. Regarding the results of our thematic analysis, while certain identified risks can be recognized as common software security threats that can be addressed with established cybersecurity best practices, others are unique to MLOps and ML-enabled systems, primarily related to data and model management. Therefore, we emphasize the need to further research and develop advanced practices for data security and continuous model monitoring.

To the best of our knowledge, this is the first study to cover the security aspects of MLOps. By adopting the MLR methodology, we were able to capture the most current information on this topic. The contributions of this paper are two-fold. First, we potentially highlight a gap in current research on AI engineering. This contribution can inform and inspire future studies of other researchers interested in MLOps. Second, we synthesize the existing body of knowledge on the theme of MLOps security, providing a categorization of the currently known risks and best practices. The resulting insights can be useful for practitioners who can use them to increase their awareness about some of the existing security threats that can affect their MLOps pipelines and thus implement current best practices to mitigate them.

2. Background

2.1. ML Systems Security

A vulnerability is defined as "*a set of conditions or behaviors that allow the violation of an explicit or implicit security policy*" [3]. Most ML systems and models are vulnerable to security attacks [4]. There is a considerable amount of research on adversarial AI, whose aim is to identify and craft input perturbations at training and/or testing time that an adversary can use to attack a model [5]. Refer to Oseni et al. [4] for a recent systematic reviews on adversarial ML.

However, models and algorithms are only a limited part of an ML system. Due to its many components, a deployed ML system exposes a broader attack surface than the training and testing of its model [6]. In addition to the models themselves, vulnerabilities can arise at other stages of development pipelines, for example, in sensors that collect data and in monitoring tools [7]. Kumar et al. [8] introduced the idea of *attacking the ML supply chain* to highlight the security threats associated with the deployment of a ML-enabled system. According to Williams et al. [9], ML systems take advantage of the same small subset of components, making any vulnerability discovery an effective means of a widespread attack. For example, because of the

large reuse within the Docker container supply chain, the discovery of one vulnerability in a parent image can lead to mass exploitation of all the child images and the systems built around them. As a result, manipulating the ML supply chain can be an effective way to change model behavior at run-time, even beyond the scope of classic adversarial tactics [9].

2.2. DevOps Security

DevOps is a methodology that promotes collaboration, communication, and integration between development and operations to narrow the gap between them [1]. The widespread adoption of DevOps is due to the gains in business value reported by industry practitioners and academic researchers [10, 11], such as the ability to deploy releases faster and more frequently [12]. However, rapid delivery practices have presented new challenges for organizations, such as ensuring security practices while maintaining DevOps agility [13]. The need to integrate security controls and practices as a priority from the beginning and throughout the DevOps lifecycle, rather than at the end, has led to the definition of the DevSecOps methodology (also known as SecDevOps and SecOps) [13].

There exist several systematic reviews and mapping studies on DevOps security [14, 13, 15, 16, 17, 18]. They focus on providing definitions, characteristics, benefits, and challenges of DevSecOps. The most recent and comprehensive systematic review was by Rajapakse et al. [19]. They present results on the challenges of DevSecOps and existing solutions in the literature categorized into four themes: *people* (i.e., issues and solutions related to knowledge, skills, collaboration among members of the DevSecOps team, and organizational culture), *practices* (i.e., integrating security practices into continuous SE practices such as CI/CD), *tools* (i.e., integrating security tools in a DevOps pipeline), and *infrastructure* (i.e., adopting DevSecOps in various types of infrastructure).

3. Methodology

Garousi et al. [20, 21, 22] showed the benefits of complementing SLRs with gray literature in software engineering research. Practitioners often lack the time or expertise to access academic publications, thus relying primarily on gray literature to share their ideas, experiences, and stay up-to-date. Next, we briefly outline the methodology used to perform our MLR. For details about the MLR protocol, please refer to the paper Appendix available on Zenodo.[1]

3.1. Search Query Definition

Initially, we curated a list of appropriate search terms for our MLR. Our process involved assembling an initial set of terms, incorporating words we had come across in the literature along with their synonyms. Subsequently, we expanded our search terms by considering keywords defined by authors in the preliminary search results. At the end of the process, we defined the following search query:

[1] https://zenodo.org/doi/10.5281/zenodo.11001417

(”MLOps” *OR* ”MLSecOps” *OR* ”SecMLOps” *OR* ”machine learning” *OR* ”ML” *OR* ”machine learning pipeline” *OR* ”AI pipeline” *OR* ”data pipeline” *OR* ”model lifecycle” *OR* ”model deployment” **)**
AND
(”security” *OR* ”threat” *OR* ”vulnerability” *OR* ”risk” *OR* ”cybersecurity” *OR* ”attack” *OR* ”SecOps” *OR* ”DevSecOps” *OR* ”SecDevOps” *OR* ”protection” *OR* ”safety” **)**

3.2. Data Collection

We used Google Scholar to retrieve the white literature and Google Search to find relevant gray literature. Preliminary analysis of titles, abstracts, and keywords from a comprehensive sample of search hits from both search engines yielded 60 potentially interesting results.

To select only relevant and high-quality items from the search results, we systematically applied a list of inclusion and exclusion criteria. Specifically, we included search results containing MLOps security risks and related best practices. We only kept search results that were articles, technical papers, PhD theses, or blog posts written in English. We excluded other types of multimedia such as podcasts and product release notes. We also excluded search results that were off-topic or contained only general information about securing the MLOps pipeline. Instead, we kept results that provided actionable insights or takeaways for securing an ML pipeline. By applying these filtering criteria, we ended up with a corpus of 13 articles.

Given the low number of selected papers, we tried to use backward and forward snowballing techniques [23] to find additional academic resources that we may have missed during the automated search; however, we were unable to select additional papers. The list of resources collected at the end of the collection process is presented in Table 1.

3.3. Data Extraction & Analysis

After carefully reading each of the selected articles, we extracted excerpts from the recommendations given and collected a raw catalog of security risks and best practices related to MLOps —by ’best practice’, we specifically mean optimal behavior that can be adopted to improve a security-related aspect of MLOps activities. Finally, to further refine our catalog, we collab-

Table 1
Results matching the inclusion criteria for the multivocal literature review focusing on security in MLOps.

Search engine	ID	Result
Google	[G1]	[24]. Network security checklist for MLOps solutions. Azure Architecture Center.
Google	[G2]	[25]. Top 7 Layers of MLOps Security | Advanced Guide
Google	[G3]	[26]. MLOps security best practices
Google	[G4]	[27]. As MLOps Hits Maturity it’s Time to Consider Cybersecurity
Google	[G5]	[28]. Why cybersecurity is critical in MLOps
Google	[G6]	[29]. 7 layers of MLOps security
Google	[G7]	[30]. Five biggest risks of AI and Machine Learning that MLOps platforms help to address
Google	[G8]	[31]. Infusing Security into MLOps
Google	[G9]	[32].The Importance of Secure Development in MLOps
Google	[G10]	[33]. MLSecOps: Defined
Google	[G11]	[34]. Adopting MLSecOps: Securing Machine Learning at Scale
Google	[G12]	[35]. Adopting MLSecOps for secure machine learning at scale
Google Scholar	[W1]	[36]. Conceptualizing the Secure Machine Learning Operations (SecMLOps) Paradigm

Table 2
The list of security risks in MLOps and best practices to counteract them. The numbers between brackets represent the support for each risk/best practice.

Category	Definition	Security Risks	Best Practices
Authentication & Authorization	Processes to control access to resources based on user identity and permissions.	– Unauthorized/ unrestricted access to data and models [G12] – Data exfiltration [G1] – Data poisoning [G3-6,G9,G11] – Model tampering, inversion, or theft [G2,G3-5,G8,G11,G12] – Malicious insiders compromising the system [G2,G3,G5,G6]	– Use service principals or managed identities for authentication [G1] – Implement access control policies and log user activity [G1,G2-4,G6,G12] – Apply zero-trust and PLoP policies to limit access [G3,G5] – Data provenance/lineage [G6] – Encrypt and sign models to prevent tampering [G11] – Implement differential privacy or federated learning [G4]
Network Security	Measures to secure network communication and isolate environments from public Internet.	– Data breaches [G9] – Adversarial attacks [G4,G5,G8-12]	– Isolate the environment using virtual networks [G1] – Use secure communication channels [G2] – Use encryption to reduce attack surface [G1-3,G6,G7] – Hash, tokenize, or mask sensitive data [G2,G4,G6] – Validate and verify input/output data [G3,G4,G11] – Implement private endpoints and firewalls [G1,G9] – Adversarial training [G10,G11]
Deployment Security	Ensuring security during model deployment.	– Vulnerabilities in deployed models [G8]	– Automate deployment [G8] – Use security testing [G3,G11] – Monitor models for performance and health [G2]
Continuous Monitoring	Continuously monitoring and ensuring model performance and behavior.	– Lack of visibility into model behavior and performance [G7] – Failure to detect model performance drops in production [G2]	– Monitor the behavior of models and their pipelines with metrics and logs [G1-5,G7,G8,G10,G11] – Use anomaly detection and alert systems [G2,G4,G5,G11] – Employ feedback loops and retraining mechanisms [G2,G4,G11]
Privacy & Ethical Guidelines Compliance	Adhering to regulations, standards, and best practices to ensure responsible lawful use of data and models.	– Non-compliance with data or model regulations [W1,G11]	– Ensure compliance with data privacy regulations [W1,G2-4,G7,G11] – Incorporate ethical guidelines into ML development [G10,G11]
Secure Development Practices	Applying secure coding practices during development.	– Code injections [G4,G5]	– Regular-update security measures [G8]
Supply Chain Security	Assessing and mitigating risks associated with tools, libraries, and dependencies used in model development and deployment.	– Vulnerabilities in tools and dependencies [G4,G5,G8,G10-12] – Lack of security testing for third-party models [G4,G5]	– Scan for known vulnerabilities in software dependencies [G8] – Test and update third-party components and models for performance, faireness and security [G4,G8,G10,G12] – Test and validate model performance, fairness, and security [G4,G8,G10]
Security Mindset and Culture	Creating a culture that prioritizes security throughout the ML development lifecycle.	– Lack of security awareness [W1]	– Train staff to foster a security-first mindset [W1,G9,G11] – Prepare for and respond effectively to security incidents [G3,G8]

oratively performed a thematic analysis of the raw collection of best practices and security risks [37].

4. Results

4.1. RQ1: Definition of MLOps Security

In our review, we could not identify a common, broadly accepted definition of MLOps security, typically referred to as SecMLOps or MLSecOps. In the white literature, we only found one definition by Zhang and Jaskolka [36] [W1] who conceptualize SecMLOps at a high level as "*the explicit consideration and integration of security within the whole MLOps life cycle to result in more secure, reliable, and trustworthy ML-based systems.*"

In the gray literature, we found a couple of definitions of MLSecOps, which appears to be the term preferred by practitioners. [G10] defines MLSecOps and also compares it against DevSecOps: "*MLSecOps refers to the integration of security practices and considerations into the ML development and deployment process. This includes ensuring the security and privacy of data*

used to train and test models, as well as protecting deployed models and the infrastructure they run on from malicious attacks. [...] MLSecOps focuses on securing machine learning models and processes, while DevSecOps focuses on securing software development and delivery processes." In [G11], MLSecOps is defined as "*implementing and managing a set of processes, tools, and best practices that are designed to secure machine learning models and the systems that support them. It aims to address the unique challenges of securing ML models at scale.*"

4.2. RQ2-3: Risks & Best Practices

In the following, we present a synthesis of risks and best practices extracted from select sources. The synthesized results of this analysis are also briefly listed in Table 2.

4.2.1. Authentication and Authorization

Authentication and authorization involve controlling access to resources and ensuring that only authorized users and applications can interact with them.

Security Risks A significant security concern related to the theme of authentication and authorization is the ***unauthorized/unrestricted access to data and models***. Concerning data, authentication and authorization issues may result in ***data exfiltration*** phenomena, which occur when unauthorized individuals or entities gain access to and extract sensitive data from a system without proper protection; similarly, problems with authentication and authorization may allow ***data poisoning*** scenarios, involving attackers that inject malicious data into training datasets, leading to incorrect patterns learned by machine learning models.

Concerning models, failing to properly ensure authentication and authorization policies may lead to ***model tampering, inversion, or theft***: these risks refer to the manipulation, reverse engineering, or unauthorized copying of machine learning models that lead to compromised functionality or intellectual property theft. *"An attacker can reverse-engineer an ML model, replicate, and exploit it for personal gains"* (G3). Moreover, *"data used to train a system can often be recovered by attackers with various techniques"* (G4).

A further risk related to this theme is represented by ***malicious insiders compromising the system***. It involves insiders with malicious intent exploiting their access privileges to compromise the security of the system and the data within it.

Best Practices Various best practices can be implemented to mitigate these risks. For instance, the ***use of service principals or managed identities for authentication***, which improves security by allowing applications and services to access resources without relying on interactive authentication. Also, ***implementing access control policies and logging user activity***: access control policies (e.g., *"direct access control (DAC — users assigned permissions) or role-based access control (RBAC — which roles can access data)"* (G6)) assign specific access rights to users or groups, limiting their permissions to only what is necessary for them to perform their tasks. Beyond authenticating users, it is also appropriate to *"continuously monitor their activity"* (G3). Another relevant best practice is to ***apply zero-trust and PLoP policies to limit access***: the zero-trust security model assumes that no entity, whether internal or external, is trusted by

default and access is granted based on need-to-know principles. Consequently, all data access requests need to be authenticated and authorized, thus reducing the risk of data breaches. Likewise, the principle of least privilege (PLoP) *"dictates that a user should have the exact access they need to perform their tasks —not more and not less."* (G3)

A further step in mitigating data security risks is ensuring that information about ***data provenance/lineage*** is available: this *"can help you audit who has accessed the data, where it came from, and how changing the data may affect downstream processes."* (G6). It is also recommended to ***encrypt and sign models to prevent tampering***, protecting them from unauthorized modification. More advanced practices to improve model security involve the implementation of ***differential privacy or federated learning***, which protect user privacy by aggregating data and knowledge while preventing individual data from being exposed in the model training process.

4.2.2. Network Security

Network security focuses on protecting communication pathways and network infrastructure within a system, ensuring that data remain confidential and secure during transit.

Security Risks Common security risks in this space include ***data breaches***, which occur when network vulnerabilities are exploited by attackers to gain unauthorized access to data or systems. Another category of risk is represented by ***adversarial attacks***, also known as 'evasion attacks.' These involve crafting inputs that are specifically designed to cause a model to make incorrect predictions. This can potentially lead to incorrect decisions or actions based on flawed output when such inputs are submitted to model instances in production systems.

Best Practices Several best practices are recommended in the literature to mitigate these risks. As a first step, it is advisable to ***isolate the environment using virtual networks*** and ***use secure communication channels***, as secure communication protocols ensure that data transmitted between components of the machine learning system remains confidential and tamper-proof. Also, it is always recommended to ***use encryption to reduce attack surface***: encrypting data as they travel across networks ensures that, even if intercepted, the data remain unreadable to unauthorized entities. This best practice should be applied not only in production settings, but since the early phases of the MLOps workflow, e.g.: *"Encrypt training data in transit and at rest by using platform-managed or customer-managed access keys"* (G1). Likewise, it is recommended to ***hash, tokenize, or mask sensitive data***: implementing such techniques helps protect sensitive information while still allowing legitimate uses —or at least it represents a good trade-off.

In addition, practitioners are advised to ***validate and verify input/output data***, which helps preventing attacks based on manipulating data inputs or stealing model outputs. Furthermore, it is important to ***implement private endpoints and firewalls***, which add an additional layer of security by restricting access to designated, trusted sources. For instance, it is advisable to forbid direct access to the models of a system: *"Making your model endpoint publicly accessible may expose unintended inferences or prediction metadata that you would rather keep private."* As a further measure to counter adversarial attacks, the literature recommends implementing

adversarial training techniques —e.g., to use *"generative models to create synthetic training data, incorporating adversarial examples into the training process, and develop robust [models] that can handle noisy inputs"* (G10).

4.2.3. Deployment Security

Deployment security focuses on ensuring that machine learning models are securely deployed, maintained, and accessible to users and systems.

Security Risks Concerning deployment security, the analyzed literature broadly refers to potential ***vulnerabilities in deployed models***, which can be exploited by attackers to compromise system integrity or misuse the model's capabilities.

Best Practices Related best practices involve, for instance, to ***automate deployment*** pipelines, leveraging pipeline orchestrators to streamline the deployment process while ensuring consistency and reducing the risk of human errors. Complementarily, it is advisable to ***use security testing***: e.g., vulnerability testing, penetration testing, threat modeling. Furthermore, it is recommended to ***monitor models for performance and health***: continuous monitoring ensures that the models deployed perform optimally and are resistant to potential problems. In general, *"It should be easy to keep an overview of how all your models are doing in production. Monitor things like data traffic, CPU usage (a spike might allude to code injection), and data drift"* (G4).

4.2.4. Continuous Monitoring

Continuous monitoring involves ongoing surveillance of machine learning models, data, and systems to identify anomalies, evaluate performance, and ensure security.

Security Risks Failing to implement an adequate monitoring system generally implicates a ***lack of visibility into model behavior and performance***, making it difficult to identify degradation in model performance or detect anomalies. Also, without an accurate monitoring system in place, there is a high risk of ***failure to detect model performance drops in production***: anomalies, such as unexpected deviations in model behavior, can go unnoticed if not proactively monitored, potentially indicating security breaches or data quality issues.

Best Practices To counter these threats, practitioners are advised to ***monitor the behavior of models and their pipelines with metrics and logs***, as regularly collecting insight into model performance helps detect anomalies. This should be done not only for production models, but also for all the tasks in the ML pipeline, indeed: *"a vulnerability introduced early in the training data will affect the model's performance down the line. "* (G5). Another recommendation is to ***use anomaly detection and alerting systems***, which enable a rapid response to potential security threats or system issues. Also, it is crucial to ***employ feedback loops and retraining mechanisms***: these ensure that models continuously learn from new data and adapt to changes

in the environment, maintaining accurate and secure predictions. Any time a deployed model starts failing, *"deploying a new version should be quick."* (G4)

4.2.5. Privacy and Ethical Guidelines Compliance

Data privacy and compliance involve adhering to regulations, ethical standards, and best practices to ensure responsible and lawful use of data.

Security Risks The main risk in this context is the ***non-compliance with data or model regulations***: failure to comply with data protection regulations, such as GDPR or CCPA, can result in legal consequences and reputational damage.

Best Practices Mitigating such risk demands a commitment to ***ensure compliance with data privacy regulations*** by design. Another crucial recommendation is to ***incorporate ethical guidelines into ML development***, implementing processes to ensure that *"models are developed and maintained according to best practices and standards [and] adhere to legal and ethical guidelines"* (G11) to account for fairness, risk of bias, explainability, and accountability.

4.2.6. Secure Development Practices

Secure development practices involve integrating security into the software development lifecycle to prevent vulnerabilities and ensure overall system security. The emphasis is on coding securely, implementing best practices, and following security guidelines during the development process itself. It involves actions such as proper input validation, avoiding known vulnerabilities, using secure coding patterns, and integrating security testing.

Security Risks A security risk related to this theme is represented, for instance, by ***code injections***, i.e., *"Carefully crafted inputs to a model [that] can cause external (malicious) code to be triggered"* (G5). Failure to implement proper input validation and sanitation techniques can expose MLOps systems to code injection attacks.

Best Practices The main best practice here involves ***regular-update security measures***, i.e., to keep security patches, libraries, and dependencies up to date, mitigating vulnerabilities and thus minimizing the risk of exploitation.

4.2.7. Supply Chain Security

Supply chain security involves assessing and mitigating the risks associated with tools, libraries, and dependencies used in the development and deployment of machine learning models. The focus is on evaluating and managing risks associated with third-party components, including libraries, frameworks, and software dependencies. This category is particularly important in an environment where organizations rely on external tools, as vulnerabilities in these tools could lead to security breaches.

Security Risks One main source of security risks related to the supply chain is represented by the presence of ***vulnerabilities in software tools and dependencies***: using outdated or vulnerable software components can introduce security weaknesses into the system. Another area of risk concerns the ***lack of security testing for third-party models***, which may also be the source of vulnerabilities that attackers can exploit.

Best Practices A first best practice to mitigate supply chain security risks is to regularly ***scan for known vulnerabilities in software dependencies***. Also, it is recommended to accurately ***test and update third-party components and models for performance, fairness, and security*** before integrating them into a system, including evaluating their architecture and potential risks. Furthermore, it is advisable to rigorously ***test and validate model performance, fairness, and security*** before deploying them to production.

4.2.8. Security Mindset and Culture

Security mindset and culture refer to collective awareness, attitudes, and behaviors within an organization that prioritize security considerations throughout the entire machine learning development and deployment lifecycle.

Security Risks The main source of risk in this area is the ***lack of security awareness***: when individuals or teams lack awareness of best practices for security and potential threats, they may inadvertently engage in actions that compromise the security of the MLOps process and its components. This could include *"careless or poorly trained staff"* (G2) who overlook security considerations during development, deployment, and monitoring.

Best Practices To mitigate this risk, the literature recommends to ***train staff to foster a security-first mindset***, establishing a culture in which security considerations are integral to every aspect of the machine learning development and deployment process. Furthermore, it is essential to ***prepare for and respond effectively to security incidents***, i.e., to develop incident response plans to handle security breaches swiftly and minimize potential damage or data exposure.

5. Discussion

Here, we answer the three research questions and discuss the implications of our findings.

5.1. A Comprehensive Definition of MLOps Security

Our review has revealed that a comprehensive and accepted definition of MLOps security is still lacking in the literature. We found that only three sources provided a definition of MLSecOps / SecMLOps, and that the two terms are used interchangeably. As such, below, we provide a definition that combines existing ones as follows: *MLSecOps / SecMLOps is the comprehensive integration of security practices throughout the entire ML pipeline. It focuses on protecting data*

privacy, securing models at scale, and ensuring the resilience of both the models and the underlying software infrastructure against malicious threats.

5.2. Risk Sources

The second research question in this study was to determine the currently known risks that affect ML-enabled systems and their development pipelines. We answer by outlining a conceptualization of the risks found based on the sources of risk. As such, we categorize potential sources of risk into three main branches: *cyberattacks*, *system/model performance degradation*, and *lack of a security culture*.

Cyberattacks Most of the risks that affect ML-enabled systems and their pipelines come from cyberattacks. In particular, our data suggest the existence of three main threat types: *unauthorized access*, *system tampering*, and *model tampering*. Attackers may act with the goal of gaining *unauthorized access* to a system, its data, or its models. In order to do so, they can exploit security issues in the authentication / authorization mechanisms of a system, in the system network, or in its deployment workflows. Often, such violations are aimed at data theft; in this regard, attackers might even exploit sophisticated strategies like model inversion (i.e., reverse engineering of ML models through which one can gain information on the data used for model training, which might contain sensitive items). Sometimes, models themselves can be the object of theft. Alternatively, unauthorized access can be exploited to compromise the availability of a system. In other cases, vulnerabilities found in code or in a project supply chain can be leveraged for *system tampering* with the goal of misusing the attacked system and the effect of jeopardizing its overall reliability. Furthermore, cyberattackers might be interested in altering the intelligent behavior of a system (*model tampering*), e.g., by replacing a model entirely or by inducing perturbation in model output by means of data poisoning techniques or adversarial attacks.

System/model performance degradation Another commonly recognized source of risk is represented by *performance degradation* phenomena, which can affect both systems and models. Such phenomena are particularly troublesome in the case of safety-critical or mission-critical systems (e.g., autonomous driving and stock prediction) whose correct operation strictly depends on the correct behavior of the ML models that power them. For instance, a sudden *drop in model performance* – e.g., due to model drift phenomena – might induce unsafe behavior in the systems relying on it.

Lack of security culture A third source of risk stems from the *lack of a security culture* in software companies adopting AI, often resulting in poor system design. For example, in the pursuit of building the best model, a lack of security awareness may cause data science teams to overlook privacy concerns and engage in the unauthorized use of sensitive information. The uncontrolled use of data and modern ML capabilities has the potential to render ML-enabled systems unethical and capable of doing harm to individuals.

5.3. General and Domain-specific Security Best Practices

By answering our third research question, we also aimed to explore the currently known MLOps best practices employed by practitioners to mitigate security risks. Upon examining the results of the thematic analysis, we observed that risks and related best practices of general validity and applicability coexist with domain-specific risks and best practices – i.e., those specific to ML-enabled systems and their development pipelines.

Common security risks and best practices Most of the risks and best practices analyzed are widely recognized in the cybersecurity domain. For instance, *'Malicious insiders compromising the system'* or *'Unauthorized / unrestricted access to sensitive data'* are commonly acknowledged cybersecurity risks that can be mitigated through best practices like *'Apply a zero-trust and PLoP policies to limit access'*, *'Isolate the environment using virtual networks'*, and *'Use encryption to reduce attack surface'*. However, in several cases, the same set of risks applies to new targets that are specific to the ML domain (e.g., large training datasets and models). Therefore, the best practices typically adopted to contrast these risks should be extended to effectively protect datasets, ML models, and related pipelines.

Domain-specific security risks and best practices On the other hand, some risks are unique to the domain of ML-enabled systems, and their mitigation requires dedicated best practices. Notable examples are cyberattacks-related risks aimed at the fraudulent manipulation of data and models: attackers may adopt advanced techniques such as *'data poisoning'* and *'adversarial attacks'* to alter model behavior. Also, they can leverage deployed or stolen models to apply reverse engineering techniques (*model inversion*) and infer sensitive data from model predictions. Best practices aimed at preventing such kinds of risks require *encrypting training data* as well as preventing the inference of sensitive data by applying *hashing, tokenization, and masking* techniques to training datasets. Similarly, to avoid exposing data to privacy and security breaches, advanced techniques like *federated learning* can be employed; this technique decentralizes model-building processes and removes the need to share or transfer local data samples, thus reducing the overall attack surface.

6. Conclusion

In this paper, we have identified and analyzed risks and best practices concerning MLOps security, based on a Multivocal Literature Review. Specifically, we have found that several security notions in MLOps are common to other cybersecurity domains. However, their awareness and application in practical ML systems engineering are still unclear and require further investigation. Moreover, we have observed that security in MLOps is largely dependent on the safe management of data and models, which calls for the definition of advanced data security and continuous model monitoring techniques. All in all, the few relevant sources identified with our systematic search suggest that our current understanding of MLOps security is still in an early stage of development. Therefore, we call for additional research in this field.

7. Acknowledgments

This work was partially supported by the project SERICS (PE00000014, CUP: H93C22000620001) under the MUR National Recovery and Resilience Plan funded by the European Union – NextGenerationEU.

References

[1] L. Bass, I. Weber, L. Zhu, DevOps: A software architect's perspective, Addison-Wesley Professional, 2015.
[2] M. Treveil, N. Omont, C. Stenac, K. Lefevre, D. Phan, J. Zentici, A. Lavoillotte, M. Miyazaki, L. Heidmann, Introducing MLOps, O'Reilly Media, Inc., 2020.
[3] A. D. Householder, G. Wassermann, A. Manion, C. King, The cert guide to coordinated vulnerability disclosure, Software Engineering Institute, Pittsburgh, PA (2017).
[4] A. Oseni, N. Moustafa, H. Janicke, P. Liu, Z. Tari, A. Vasilakos, Security and privacy for artificial intelligence: Opportunities and challenges, arXiv preprint arXiv:2102.04661 (2021).
[5] B. Biggio, F. Roli, Wild patterns: Ten years after the rise of adversarial machine learning, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 2154–2156.
[6] J. M. Spring, A. Galyardt, A. D. Householder, N. VanHoudnos, On managing vulnerabilities in ai/ml systems, in: Proceedings of the New Security Paradigms Workshop 2020, NSPW '20, Association for Computing Machinery, New York, NY, USA, 2021, p. 111–126. doi:10.1145/3442167.3442177.
[7] Y. Hu, W. Kuang, Z. Qin, K. Li, J. Zhang, Y. Gao, W. Li, K. Li, Artificial intelligence security: Threats and countermeasures, ACM Comput. Surv. 55 (2021). doi:10.1145/3487890.
[8] R. S. S. Kumar, D. O. Brien, K. Albert, S. Viljöen, J. Snover, Failure modes in machine learning systems, arXiv preprint arXiv:1911.11034 (2019).
[9] D. Williams, C. Clark, R. McGahan, B. Potteiger, D. Cohen, P. Musau, Discovery of ai/ml supply chain vulnerabilities within automotive cyber-physical systems, in: 2022 IEEE International Conference on Assured Autonomy (ICAA), 2022, pp. 93–96. doi:10.1109/ICAA52185.2022.00020.
[10] A. Mann, M. S. A. Brown, N. Kersten, State of devops report, Accessed: Jul 2018 (2019).
[11] B. Fitzgerald, K.-J. Stol, Continuous software engineering: A roadmap and agenda, Journal of Systems and Software 123 (2017) 176–189.
[12] L. Riungu-Kalliosaari, S. Mäkinen, L. E. Lwakatare, J. Tiihonen, T. Männistö, Devops adoption benefits and challenges in practice: A case study, in: P. Abrahamsson, A. Jedlitschka, A. Nguyen Duc, M. Felderer, S. Amasaki, T. Mikkonen (Eds.), Product-Focused Software Process Improvement, Springer International Publishing, Cham, 2016, pp. 590–597.
[13] H. Myrbakken, R. Colomo-Palacios, Devsecops: A multivocal literature review, in: A. Mas, A. Mesquida, R. V. O'Connor, T. Rout, A. Dorling (Eds.), Software Process Improvement and Capability Determination, Springer International Publishing, Cham, 2017, pp. 17–29.
[14] V. Mohan, L. B. Othmane, Secdevops: Is it a marketing buzzword?-mapping research on

security in devops, in: 2016 11th international conference on availability, reliability and security (ARES), IEEE, 2016, pp. 542–547.

[15] L. Prates, J. Faustino, M. Silva, R. Pereira, Devsecops metrics, in: S. Wrycza, J. Maślankowski (Eds.), Information Systems: Research, Development, Applications, Education, Springer International Publishing, Cham, 2019, pp. 77–90.

[16] M. Sánchez-Gordón, R. Colomo-Palacios, Security as culture: a systematic literature review of devsecops, in: Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops, 2020, pp. 266–269.

[17] R. Mao, H. Zhang, Q. Dai, H. Huang, G. Rong, H. Shen, L. Chen, K. Lu, Preliminary findings about devsecops from grey literature, in: 2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS), 2020, pp. 450–457. doi:10.1109/QRS51102.2020.00064.

[18] S. Rafi, W. Yu, M. A. Akbar, A. Alsanad, A. Gumaei, Prioritization based taxonomy of devops security challenges using promethee, IEEE Access 8 (2020) 105426–105446. doi:10.1109/ACCESS.2020.2998819.

[19] R. N. Rajapakse, M. Zahedi, M. A. Babar, H. Shen, Challenges and solutions when adopting devsecops: A systematic review, Information and Software Technology 141 (2022) 106700. URL: https://www.sciencedirect.com/science/article/pii/S0950584921001543. doi:https://doi.org/10.1016/j.infsof.2021.106700.

[20] V. Garousi, M. Felderer, M. V. Mäntylä, The need for multivocal literature reviews in software engineering: Complementing systematic literature reviews with grey literature, in: Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, EASE '16, Association for Computing Machinery, New York, NY, USA, 2016. doi:10.1145/2915970.2916008.

[21] V. Garousi, M. Felderer, M. V. Mäntylä, Guidelines for including grey literature and conducting multivocal literature reviews in software engineering, Information and Software Technology 106 (2019) 101–121. doi:https://doi.org/10.1016/j.infsof.2018.09.006.

[22] V. Garousi, M. Felderer, M. V. Mäntylä, A. Rainer, Benefitting from the Grey Literature in Software Engineering Research, Springer International Publishing, Cham, 2020, pp. 385–413. doi:10.1007/978-3-030-32489-6_14.

[23] C. Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: M. J. Shepperd, T. Hall, I. Myrtveit (Eds.), 18th International Conference on Evaluation and Assessment in Software Engineering, EASE '14, London, England, United Kingdom, May 13-14, 2014, ACM, 2014, pp. 38:1–38:10. doi:10.1145/2601248.2601268.

[24] G. Wang, G. Moore, Network security checklist for MLOps solutions, https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/network-security-mlops, 2023.

[25] N. Singh Gill, Top 7 Layers of MLOps Security | Advanced Guide, https://www.xenonstack.com/blog/mlops-security-layers, 2023.

[26] MLOps Security Best practices, https://www.trendmicro.com/en_id/devops/23/b/mlops-security-best-practices.html, 2023.

[27] A. Dutrée, As MLOps Hits Maturity it's Time to Consider Cybersecurity, https://towardsdatascience.com/as-mlops-hits-maturity-its-time-to-consider-cybersecurity-ebd45e350532, 2022.

[28] G. Ollmann, Why Cybersecurity is Critical in MLOps, https://www.securityinfowatch.com/-cybersecurity/article/53061904/why-cybersecurity-is-critical-in-mlops, 2023.
[29] D. Linkov, 7 Layers of MLOps Security, https://denyslinkov.medium.com/7-layers-of-mlops-security-5bfd87eea928, 2021.
[30] S. Mohanty, Five biggest risks of AI and Machine Learning that MLOps platforms help to address, https://community.nasscom.in/communities/data-science-ai-community/five-biggest-risks-ai-and-machine-learning-mlops-platforms, 2022.
[31] H. Anderson, Infusing Security into MLOps, https://www.robustintelligence.com/blog-posts/mlsecops-infusing-security-into-mlops, 2023.
[32] A. Poulton, The Importance of Secure Development in MLOps, https://www.equalex-perts.com/blog/tech-focus/mlops-security-tips-development/, 2022.
[33] What is MLSecOps, https://mlsecops.com/what-is-mlsecops, 2023.
[34] B. Ghosh, Adopting MLSecOps: Securing Machine Learning at Scale, https://medium.com/@bijit211987/adopting-mlsecops-securing-machine-learning-at-scale-1a5647d01a64, 2023.
[35] A. Saucedo, Adopting MLSecOps for secure machine learning at scale, https://venture-beat.com/datadecisionmakers/adopting-mlsecops-for-secure-machine-learning-at-scale/, 2022.
[36] X. Zhang, J. Jaskolka, Conceptualizing the Secure Machine Learning Operations (SecM-LOps) Paradigm, in: 2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS), IEEE, Guangzhou, China, 2022, pp. 127–138. doi:10.1109/QRS57517.2022.00023.
[37] V. Clarke, V. Braun, Thematic analysis, The journal of positive psychology 12 (2017) 297–298.