

# Monitoring Machine Learning Systems from the Point of View of AI Ethics

Kai-Kristian Kemell<sup>1,\*</sup>, Jukka K. Nurminen<sup>1</sup> and Ville Vakkuri<sup>2</sup>

<sup>1</sup>University of Helsinki, Department of Computer Science, Yliopistonkatu 3, 00014 University of Helsinki, Finland

<sup>2</sup>School of Marketing and Communication, University of Vaasa, Wolffintie 32 FI-65200 Vaasa PL 700, Vaasa, FI-65101, Finland

## Abstract

The practical implementation of AI ethics remains a challenge. Guidelines and principles are numerous but converting them into practice appears difficult for organizations developing ML systems. It is argued that bringing AI ethics closer to software engineering practice could help. In this regard, monitoring of ML systems and metrics related to ethics could be one way of making ethics more tangible. While various existing papers discuss technical approaches to, for example, monitoring fairness, a more holistic view on monitoring AI ethics is lacking, as is discussion on MLOps and ethics. In this paper, we discuss AI ethics from the point of view of monitoring, building on existing research from AI ethics, software engineering, and machine learning, to propose a typology of metrics for monitoring ML systems during their operational lives. We then discuss monitoring ML systems from the point of view of AI ethics by building on this typology and using the Ethics Guidelines for Trustworthy AI (AI HLEG) as a framework to illustrate what monitoring AI ethics might mean in practice. In doing so, we highlight that (a) some issues related to AI ethics are hardly unique to AI ethics and are something frequently tackled in ML monitoring, (b) though AI ethics involves many high-level design decisions to be made early on in the development of a system, there are still various aspects of AI ethics that may be monitored. Overall, this paper presents initial discussion on the topic in hopes of encouraging further studies on it.

## Keywords

AI ethics, monitoring, metrics, software engineering, ML development, ethical guidelines, MLOps

## 1. Introduction

While AI ethics has become a prominent topic of discussion that companies developing AI systems are also increasingly aware of, AI ethics remains challenging in practice for various reasons. Guidelines and principles, which are often utilized to approach AI ethics, are seen as ineffective [1], and indeed seem to have little impact on practice [2, 3]. AI ethics overall is still difficult to approach, and amidst the numerous guidelines and principles [4, 5], can appear fuzzy to practitioners who may find it difficult to define exactly *what* it is they are trying to tackle. It has been argued that one problem in this regard is that AI ethics is distant from Software Engineering (SE) practice, and that bringing AI ethics closer to the typical work of software engineers might help [6, 7].


---

7th Conference on Technology Ethics (TETHICS2024), November 6–7, 2024, Tampere, Finland

\*Corresponding author.

✉ kai-kristian.kemell@helsinki.fi (K. Kemell); jukka.k.nurminen@helsinki.fi (J. K. Nurminen); ville.vakkuri@uwasa.fi (V. Vakkuri)

ORCID 0000-0002-0225-4560 (K. Kemell); 0000-0001-5083-1927 (J. K. Nurminen); 0000-0002-1550-1110 (V. Vakkuri)

 © 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The operational life of AI systems remains a less explored point of view in AI ethics [8]. Conversely, SE has increasingly begun to emphasize continuity in SE through what has become known as continuous SE [9]. Especially DevOps, a well-known continuous SE approach, has made *operations* (or the operational life of the system) an integral part of the SE process [9]. Rarely does active development of a system stop after its initial release anymore, as new features and new versions of the system are continuously deployed, especially with software increasingly offered as a service [10]. This has also come to be the case for ML systems, especially with MLOps [11] extending DevOps principles into ML development. Whereas DevOps is a portmanteau of development and operations, MLOps is one of Machine Learning (ML) and operations.

However, few AI ethics papers discuss continuous SE, MLOps, or even just monitoring of ML systems, outside ML papers discussing technical solutions to ethics-related issues. For example, and primarily, there are ML papers discussing monitoring aspects of algorithmic fairness (e.g., [12]) and various safety and robustness aspects (which are certainly concerns outside the specific point of view of ethics as well). Other types of AI ethics papers, on the other hand, seldom discuss monitoring. To this end, Lu et al. [13] posit "there is a strong desire for continuously monitoring and validating AI systems post deployment for ethical requirements but current operation practices provide limited guidance". We also consider this a relevant gap as monitoring, and especially the metrics used to monitor ML systems, are one way in which AI ethics may very tangibly manifest in practice, making it more approachable to practitioners struggling to implement it. Some existing papers have also begun to explore MLOps in relation to ethics, using the concept *sustainable MLOps* (e.g., [14]) and presenting a link between MLOps and AI ethics.

Thus, in this paper, we discuss AI ethics from the point of view of monitoring ML systems, to build on this nascent topic of study. This is a conceptual paper that aims to encourage further studies into ML system monitoring and MLOps from the point of view of AI ethics. Based on existing literature, we (1) propose a typology for categorizing metrics for monitoring ML systems during their operational life, and include examples of various metrics for each category, in order to provide a framework for future studies, and (2) discuss AI ethics from the point of view of monitoring and metrics, using this typology and the Ethics Guidelines for Trustworthy AI [15] as frameworks. This is a conceptual paper that takes on a broad perspective on the topic.

## 2. Background

In this section, we discuss relevant literature. In Section 2.1, we discuss monitoring in SE and ML systems. In Section 2.2, we discuss MLOps. In Section 2.3, we discuss AI ethics.

### 2.1. Monitoring in Software Engineering and Machine Learning Systems

Software monitoring has historically focused largely on quality, specifically from the point of view of errors and stability, and especially in security-critical systems [16]. This is a view on monitoring still often adopted today (e.g., [17]). Schroder & Schulz [17] define monitoring as "*a process involving the end-to-end extraction, analysis, and interpretation of metrics of an object under observation.*" This definition builds on an earlier paper of Kitchenham & Walker [18] who

discuss the process of interpreting the results of monitoring as follows: (1) identification of abnormal values, (2) determination of possible causes, and (3) possible corrective actions.

While ensuring error-free operation remains an important aspect of monitoring, it can be considered a narrow view on monitoring today. Users today expect systems to be continuously developed past their initial release, with more value being added through continuous deployment of updates [10]. Indeed, continuous SE, and especially DevOps, is now the de facto industry standard [19]. From this follows that monitoring is also focused on *improving* the software based on data, instead of simply maintaining it. Rodríguez et al. [10] highlight this shift: *“the main objective of continuous monitoring is to constantly monitor and measure both business indicators and infrastructure-related metrics in order to facilitate and improve business and technical decision-making.”* Overall, SE has recently drawn from the lean startup [20] philosophy that closely links business aspects with software development [9], focusing on collecting data to determine user needs in contexts where they may not be readily apparent (vs. commissioned projects).

As for ML systems, ML system monitoring in and of itself is not a novel topic, with numerous studies discussing various aspects of ML system monitoring. ML systems differ from traditional systems with the addition of ML components. These ML model(s) have to be monitored [17]. Moreover, ML system monitoring places much emphasis on data-related metrics [17], as (lots of) data is typically used to train the ML models. Overall, many of the studies discussing ML system monitoring focus on issues unique or particularly relevant for ML systems, such as concept drift and data drift (see, e.g., [21]).

ML systems also pose unique challenges from the point of view of quality through model accuracy, whereas conventional software merely needs to operate in an error-free manner at its simplest. Though the system may be functioning correctly, the ML model may become less accurate over time due to concept and/or data drift resulting from changes to the context the system operates in [22], necessitating the deployment of new, updated models [11]. When done in a continuous manner, this referred to as MLOps, which we discuss next. However, few studies take on a more holistic view to ML system monitoring as we do in this paper. One example of a more general paper is that of Naser & Alavi [23] who provide an extensive list of error metrics and performance fitness indicators for ML systems.

## 2.2. MLOps

MLOps has been characterized as DevOps for ML [11]. It refers to the application of continuous SE practices, namely the ones present in DevOps, to ML system development. MLOps is largely concerned with improving collaboration between data scientists and other ML developers and the rest of the development team [19], much like DevOps focuses on collaboration between the development and operations teams [9]. Indeed, many of SE challenges in ML development stem from a lack communication and collaboration issues between the ML developers and the rest of the development team [24, 25].

Monitoring is also discussed in the context of MLOps, with a focus on automating model monitoring [11]. This is discussed in relation to, e.g., tooling and processes for measuring model degradation in deployed models [11]. Mäkinen et al. [22] consider monitoring in MLOps to include *“automation and monitoring at all steps of ML system development and deployment, including integration, testing, releasing, deployment and infrastructure management.”*

### 2.3. AI Ethics

The primary motivation behind this paper is to explore ML system monitoring from the point of view of AI ethics. AI ethics is a long-standing area of research that has recently been highly active following major practical advances in ML in the past decade [26]. Companies, researchers, and governmental actors alike have begun to show interest in AI ethics. This interest is perhaps best highlighted through the sheer number of guidelines for ethical AI that have been proposed in recent years (see Jobin et al. [4] and Hagendorff [5] for reviews of guidelines).

These guidelines typically approach ethical issues through *principles* that ethical AI should adhere to. Such principles have become a common way of conceptualizing AI ethics. For example, one commonly discussed principle is fairness, which is concerned with issues such as bias, diversity, and equality [4]. Fairness is also the focus of various ML tools [27, 28]. However, abstract principles are challenging to implement in practice [1], and AI ethics is a field characterized by a gap between research and practice, as companies seem to struggle to implement AI ethics in practice [2, 3] (or are not interested in doing so [29]).

Though various tools related to AI ethics have been introduced [27], their adoption seems to not be widespread [2, 3, 29]. Some recent studies have proposed SE methods for implementing AI ethics (e.g., ECCOLA [7], and RE4AI [30]), which take on a more holistic view of AI ethics, looking past individual ethical principles. However, arguably much still needs to be done to make AI ethics practical [7], and empirical studies in AI ethics are still sorely lacking [28]. This paper aims to contribute towards making AI ethics more practical.

## 3. ML System Monitoring: A Typology

In this section, we propose a typology of metrics for ML system monitoring, focusing specifically on systems that are already operational. There are various other metrics that may be relevant while training the initial models for the system, for example, and while some of these (types of) metrics may also be relevant for these purposes, our focus is on metrics that are relevant for operational ML systems. To this end, we also focus on systems where new ML model versions are deployed, possibly in a continuous manner utilizing MLOps. We propose the following seven categories of metrics, which are further discussed in Sections 3.1 to 3.7:

1. Data Metrics
2. ML Model Metrics
3. System/Infrastructure Metrics
4. Process Metrics
5. Business Metrics
6. User Metrics
7. Domain-specific Metrics

This typology takes on a holistic point of view to ML system monitoring. There are various ways to categorize metrics in the field of software engineering. In existing literature, metrics are typically discussed in specific contexts. For example, there are papers discussing DevOps metrics [31, 32], software startup metrics [33], error metrics and "performance fitness indicators"

for ML systems [23], and user experience [34]. In the future, we would also like to see such papers on ethics in software engineering, and ML development specifically. On the other hand, a more general view on metrics such as what is presented here is more difficult to come by, as it is arguably less useful in terms of practice due to how general it is as a result of its vast scope. By focusing on specific contexts such as DevOps, it is possible to have a clearer goal for the metrics presented in a paper or typology, resulting in more detail, and thus practical relevance.

Our goal is to highlight how metrics related to ethics can be found across the different categories, for which purposes such a more generic typology is useful for illustrative purposes. To this end, this paper builds on a wide variety of existing literature on more specific categories of metrics. One general way of approaching software metrics found in extant literature is to consider product, process, and resource metrics [35]. This typology includes all of these with some more granularity, given the ML context (data, ML model, system/infrastructure for product, etc.), while also building on other more recent literature on metrics in software engineering.

Literature on ML system monitoring typically focuses on data metrics and model metrics, as these are the types of metrics most focused on what separates ML systems from other software systems: the ML models and their training data. We include five other categories of metrics that are more generic, as these are *also* relevant for ML systems, given that much of ML development is ultimately still just software development [36]).

Yet this typology is **not** exhaustive, nor do we aim to provide an exhaustive list of metrics for any of the categories. As Fenton & Neil [37] point out, there were already thousands of SE metrics discussed in extant literature by the year 2000, making such a goal unrealistic. We simply wish to present a typology that could be utilized to categorize relevant metrics, while providing *examples* of metrics for each category, and to then use this as a starting point for discussing metrics and monitoring in relation to *AI ethics* specifically.

Finally, it should also be noted that the frequency of monitoring metrics varies between (types of) metrics. Some metrics are continuously monitored in real-time during run-time, such as ones measuring the live performance of the system (compute, uptime, etc.), while others may be measured on a regular basis but not in real-time, such as some model and data metrics that may be relevant to measure every time a new model version is trained or deployed. To this end, some of the metrics we discuss are not necessarily *continuously* measured during run-time, but are nonetheless measured occasionally during the operational life of an ML system.

### 3.1. Data Metrics

Given that ML systems are significantly data-driven, literature on ML system monitoring places much emphasis on data metrics. The performance of ML models is highly dependent on the quality and relevance of the data they are trained on, and the way data is processed and managed also plays a notable role in ML development. "Garbage in, garbage out" is a phrase often uttered in relation to ML systems on the relation of training data and model output quality.

This category focuses on metrics related to the data utilized by the system, and specifically *input* data. While output data metrics are important, we consider them model metrics in this typology as they reflect the attributes of an ML model. Thus, examples of data metrics include:

- Data quality and integrity metrics (such as accuracy, consistency, completeness, and timeliness, etc.).
- Data preprocessing metrics (feature extraction, normalization, imputation, etc.).
- Data diversity metrics.
- Data privacy metrics (such as the percentage of data anonymized, what is stored, where, for how long, etc.).
- Data usage metrics (such as which data is ingested and processed, data velocity, etc.).
- Data drift detection metrics (Kullback-Leibler divergence, Chi-squared test, etc.).

### 3.2. ML Model Metrics

ML model metrics measure attributes related to the ML models in an ML system, such as performance or accuracy and fairness. The very conventional model metrics are focused on accuracy, such as error rates, though more sophisticated ones have been adopted over time (e.g., precision and recall for classification tasks). Monitoring ML model performance is important as data and/or concept drift can degrade model performance over time. Data drift refers to the "gradual change in input data that impacts ML model performance" [38], while concept drift refers to a situation where the "statistical properties of the target variable, which the model is trying to predict, change over time in unforeseen ways" [39]. Examples of model metrics include:

- Numerous performance metrics (such as accuracy, precision, recall, F1 score for binary and multiclass classification; Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for regression tasks, etc.; including also subtypes such as probabilistic metrics like log loss and brier score, as well as temporal metrics for concept drift detection.) (*Naser & Alavi [23] provide an extensive review of such metrics.*)
- Fairness metrics (demographic parity, equal opportunity, equalized odds, etc.).
- Training metrics (time, computational resources, etc.).
- Model complexity (number of parameters, size, depth, etc.).
- Versioning (version, training data (set(s)) used, etc.).
- Interpretability and explainability metrics (LIME, counterfactual explanations, feature importance, etc.).

### 3.3. System/Infrastructure Metrics

System/infrastructure metrics evaluate the overall health and performance of the hardware and software infrastructure that hosts and executes the machine learning components. An ML model is ultimately just one part of a larger system [17]. System metrics are very conventional metrics in software run-time monitoring, although the inclusion of ML components both adds new metrics and may change the relative importance of existing ones. For example, ML components may encourage the monitoring of model inference time, data storage requirements, and other ML-specific infrastructure characteristics. Types of system/infrastructure metrics include:

- Hardware metrics (such as CPU/GPU/TPU usage, memory utilization, disk I/O, network latency/bandwidth, etc.).



- Software metrics (such as system load, application response time, error rates, model inference time, end-to-end latency, etc.).
- System health metrics (such as uptime/downtime, availability, failure rate, etc.).
- API metrics (such as response time, request rate, etc., if exposing ML models through an API, or for APIs utilized by the system itself).
- Environmental impact metrics (energy efficiency, carbon footprint, e-waste, etc.).
- Cybersecurity metrics (*though such metrics may also be relevant for model and data categories in ML systems*).

### 3.4. Process Metrics

In software engineering, various metrics are utilized to measure attributes related to the software development processes. These include metrics related to the development approach (e.g., DevOps metrics [31, 32]) and metrics related to the performance of individual developers (e.g., many recent studies have explored the productivity impact of generative AI tools in software development). Ultimately, much of ML system development is still software development [36], making many of these metrics also relevant for ML system development. In ML development contexts, these metrics also include metrics related to the ML workflows. Examples of process metrics include:

- Development metrics (such as code churn, code quality metrics, defect density, etc.).
- Project management metrics (such as cycle time, lead time, story points completed, burn-up/burn-down charts, requirements engineering metrics related to user stories, etc.).
- Operations metrics (such as (model) deployment frequency, mean time to recovery, model retraining frequency, model rollback rate, model deployment success rate, etc.).
- Team metrics (such as team velocity, turnover rate, employee satisfaction, number of meetings, code reviews, etc.).

### 3.5. Business Metrics

In this context, business metrics aim to measure the economic and strategic performance of ML systems. These metrics often capture the alignment of the ML system with business objectives, customer satisfaction, and financial performance, and thus provide valuable insight for decision-making at the strategic and operational levels. The importance of business aspects in SE is well-acknowledged especially in the context of continuous SE [9, 10]. While these are largely generic business metrics (e.g., return-on-investment), some ML-specific metrics can still be identified. These should still be related to the system at hand, however, rather than the overall business of the organization, although such a distinction can be difficult to make in a situation where the one system *is* the entire business of a small software company, for example. For generic business metrics for SE, and specifically for software startups, Kemell et al. [33] provide an extensive list. Examples of business metrics include:

- Financial metrics (such as ROI, development cost, revenue generated, model training, deployment, and maintenance costs, data acquisition and storage costs, etc.).

- System impacts (such as process efficiency improvement, cost savings, employee productivity, etc., either internally or in external organizations using the system).

Many key business metrics in SE are arguably related to users. However, in this typology, we feel that it is prudent to separate user metrics into their own category (Section 3.6).

### **3.6. User Metrics**

User metrics measure the interactions between the system and its users. Current SE approaches often emphasize the importance of users, with Agile placing emphasis on user involvement [40] and continuous SE involving the collection of user data to further improve the system in a data-driven manner (e.g., through A/B testing [41]). Understanding users is important in order to ensure that the system meets their needs and provides value for them [20]. Thus, user metrics are usually closely related to business aspects in practice [9]. User Experience (UX) is also a well-established research area. A large list of user metrics can be found in Kemell et al. [33]. Rodden et al. [34] discuss measuring UX overall, while, for example, Arifin et al. [42] discuss measuring UX in the more specific contexts of augmented reality applications. Moreover, in terms of user conversion (visitor to paid user), various so-called funnels are utilized in business in practice, with various related metrics being utilized. Examples of user metrics include:

- Behavioral metrics (such as use frequency, session length, user engagement, various metrics measuring how a system is used during sessions, etc.).
- User satisfaction metrics (such as user satisfaction score, usability scale, feedback, reviews, social media posts, etc.).
- Impact metrics (conversion rate, retention rate, etc.).

### **3.7. Domain-specific Metrics**

Domain-specific metrics not directly related to any of the preceding categories may be found in various domain. For example, in the medical field, data related to patients and treatment outcomes may be of interest in ML development as well. Organizations developing ML systems should be aware of such metrics for their domain. However, given the diversity of such metrics, we do not discuss them further in this paper past acknowledging them in this typology due to their potential importance for practice.

## **4. Monitoring Ethical Aspects in ML systems**

In this section, we leverage the typology of metrics presented in Section 3 to discuss the monitoring of ethics in ML systems. We utilize the Ethics Guidelines for Trustworthy AI [15] (AI HLEG) and their seven requirements as a framework for this discussion, as we look at what kinds of monitoring these requirements might necessitate in practice. This section is split into seven subsections based on the seven requirements of HLEG AI. The individual requirements are elaborated on in their respective subsections. The discussion here is hardly exhaustive and is simply meant to illustrate on a very general level how AI ethics principles (or in this case requirements) might manifest in monitoring and metrics.



#### **4.1. Monitoring the Requirement of Human Agency and Oversight**

This requirement comprises fundamental rights, human agency, and human oversight. As for fundamental rights, an assessment of an ML system's fundamental rights impact "should be done prior to the system's development" [15]. No clear avenues for monitoring are indicated. Human agency posits that users "should be able to make informed autonomous decisions regarding AI systems" and "not to be subject to a decision based solely on automated processing when this produces legal effects on users or similarly significantly affects them" [15], and is also a design-level issue first and foremost. This is also closely related to the GDPR [15].

Human oversight, however, presents clear considerations for monitoring. Human oversight involves human intervention, including, for example, the ability to decide when and how to use the system in different contexts, to oversee its activity in a very broad sense, and to override decisions made by the system [15]. Monitoring can play a key role in understanding when human intervention is required. Certain metrics can be used to raise alerts, pointing to a need for human intervention of varying kinds and of varying urgency. From the point of view of ethics in particular, this may encompass fairness drift [43] for example, which may indicate a need for model retraining in the near future, but may not be a crucial alert necessitating immediate emergency action. Such metrics may be found across multiple categories in our typology, and especially in model metrics.

#### **4.2. Monitoring the Requirement of Technical Robustness and Safety**

The requirement of technical robustness encompasses resilience to attack and security, fallback plan and general safety, accuracy, and reliability and reproducibility. Cybersecurity is an issue for ML systems like all software systems [15]. Cybersecurity is not a concern unique to AI ethics and is (or should be) a well-established concern in organizations. Various existing papers discuss monitoring ML systems in relation to cybersecurity. Attacks against ML systems can take on forms not seen against conventional software, such as attacks against data (data poisoning, etc.) or the ML model (model leakage, etc.) [15]

Fallback plan and general safety posits that AI systems should be able to deal with problems through a fallback plan, by, e.g., switching from statistical or rule-based procedure, or alerting a human operator before continuing. Such safety measures should be proportionate to the potential risk posed by the system. [15] Monitoring is necessary for determining when such actions are needed, by defining through metrics what such problems could be. Such safety concerns are also not unique to AI ethics.

Accuracy is another conventional concern for ML developers discussed in a large number of existing papers, and for which various metrics exist. AI ethics (and in this case the AI HLEG) simply places additional emphasis on being able to mitigate the risk or harm posed by inaccuracies and that the system is able to indicate how likely errors are (tying to transparency) [15]. Reliability and reproducibility are related to accuracy, with reliability referring to a system working "properly with a range of inputs and in a range of situations" and reproducibility referring to "whether an AI experiment exhibits the same behavior when repeated under the same conditions" [15].

### **4.3. Monitoring the Requirement of Privacy and Data Governance**

The requirement of privacy and data governance encompasses privacy and data protection, quality and integrity of data, and access to data [15]. Privacy and data protection is more focused on design-level decisions (training data sets used, etc.) and establishing the relevant protocols than continuous monitoring. Quality and integrity of data necessitate more continuous monitoring in the context of self-learning systems. Otherwise, metrics may be checked when any changes happen, for example, upon acquiring new training data. Access to data is an aspect of privacy and data governance that can be continuously monitored (logged) aside from establishing the relevant protocols.

### **4.4. Monitoring the Requirement of Transparency**

The requirement of transparency encompasses traceability, explainability, and communication [15]. Traceability posits that "the data sets and the processes that yield the AI system's decision, including those of data gathering and data labeling as well as the algorithms used, should be documented to the best possible standard". In AI HLEG [15], these are presented as project documentation issues, aside from the reference to model outputs that ties to explainability. Explainability "concerns the ability to explain both the technical processes of an AI system and the related human decisions" [15]. In AI ethics literature, explainability is often seen as a trade-off, where ML approaches with higher accuracy are seen to generally result in less explainability, although this is an issue primarily relevant when training ML models. Ensuring and evaluating explainability past deployment may be done via user collaboration, for which Hoffman et al. [44] extensively discuss different approaches. However, this is not a continuous monitoring issue but rather something to evaluate when a model is deployed.

Transparency also encompasses communication in AI HLEG, which includes disclaimers for informing users that they are interacting with an AI, as well as communicating the capabilities and limitations of the system (accuracy, etc.) to relevant stakeholders [15]. Out of these two aspects, the latter ties to monitoring in that such disclaimers may need to be updated with metrics that reflect the evolving capabilities and limitations of the system as it is continuously developed. For the most part, however, these are issues related to the organization and its communication rather than the ML system.

### **4.5. Monitoring the Requirement of Diversity, Non-Discrimination and Fairness**

The requirement of diversity, non-discrimination, and fairness encompasses avoidance of unfair bias, accessibility and universal design, and stakeholder participation. Extant literature extensively discusses unfair bias from the point of view of data and model outputs, and some studies in relation to (continuous) monitoring of fairness also exist [12, 43]. Fairness is indeed one of the most explored aspects of AI ethics [4] and is an issue businesses are also well aware of [45], although this hardly means that it is an easily solved one, given the various practical challenges associated with detecting and tackling it.

Accessibility and universal design, like the name implies, is more related to design decisions. However, monitoring may be relevant to ensure that the system is indeed usable by different

groups of users, in addition to user feedback. Indeed, in terms of user feedback (stakeholder participation), AI HLEG considers it beneficial to "solicit regular feedback even after deployment and set up longer term mechanisms for stakeholder participation" [15]. Continuing to develop the system based on user data (though not necessarily direct, explicit feedback) is something continuous SE generally advocates. It could be argued that continuous SE in this regard supports AI ethics, although AI ethics places more emphasis on soliciting direct feedback rather than simply utilizing system use data. Some existing practices like user exit surveys or customer satisfaction surveys may also contribute in this regard.

#### **4.6. Monitoring the Requirement of Societal and Environmental Well-Being**

The requirement of societal and environmental well-being encompasses sustainable and environmentally friendly AI, social impact, and society and democracy. Sustainable and environmentally friendly AI places emphasis on environmental friendliness, encompassing "development, deployment and use process, as well as its entire supply chain" [15]. Resources used on training models and monitoring of environmental impacts during deployment are both relevant for monitoring. On the other hand, social impact and society and democracy are more challenging from the point of view of metrics and monitoring, and are arguably more related to design decisions and decisions related to how the system is utilized in practice (where, by whom, etc.). While AI HLEG posits that, in terms of social impact, the effects of ML systems "should be carefully monitored" [15], finding relevant metrics may be challenging. User feedback and monitoring social media response, etc. may provide some indication of the wider impacts of an ML system, in addition to other types of user data.

#### **4.7. Monitoring the Requirement of Accountability**

The requirement of accountability encompasses auditability, minimisation and reporting of negative impacts, trade-offs, and redress. Auditability "entails the enablement of the assessment of algorithms, data and design processes", for internal and external auditors [15]. Various metrics can support auditability in providing attributes to assess. Minimisation and reporting of negative impacts may involve monitoring to detect any relevant negative impacts during the operational life of the system, but like social impacts, may be difficult to measure. Moreover, the AI HLEG [15] places more emphasis on reporting made by other parties than organizations operating the ML systems (whistle-blowing, trade unions, etc.).

In placing emphasis on transparency, AI ethics (and AI HLEG [15]) also includes documenting trade-offs and making them in an informed manner. However, these can be considered largely high-level decisions, rather than something to repeatedly measure or continuously monitor, although if experimenting with different model training approaches or training data, relevant trade-offs may be more regular. Finally, redress refers to organizational processes related to enabling users to seek redress, and while they may provide some relevant data on issues with the system after the fact, they are hardly primarily monitoring issues.

## 5. Discussion

In this section, we discuss the implications of this typology (Section 3) and our example of how one set of ethical guidelines [15] might be relevant for monitoring (Section 4). We also highlight issues that future research could look into, or that practitioners might consider of interest.

**Relevant literature is diverse.** This is a potential challenge for any researcher looking to conduct studies on this topic, and one limitation for this paper. For example, various technical papers include discussion on issues related to AI ethics, such as safety, fairness, and explainability, but they may not explicitly discuss ethical aspects at all and may not use concepts typically seen in AI ethics literature. Similarly, monitoring related to data aspects may be discussed in literature not at all related to AI or ML (big data, etc.). Thus, there is already much potentially relevant research that is simply framed differently.

**Ethics is not just numbers.** We wish to highlight that, though AI ethics may ultimately manifest in practice through metrics to monitor, ethics should not be reduced to just numbers. As the AI HLEG guidelines [15] also highlight, ethics involves various trade-offs and design decisions that are made during the design and development of an ML system. The selection of metrics themselves may also include trade-offs when, for example, deciding between different fairness metrics. There is no one-size fits all in ethics, and there should also be ethical decision-making involved in the process of selecting the metrics used to monitor ethical aspects.

To this end, however, **not everything about AI ethics necessitates monitoring.** Not all ethical principles (or requirements in the case of AI HLEG) directly manifest as system features [8], let alone ones that require monitoring. AI ethics also involves various design-level decisions made early on in the development of a system [15].

**Some existing concerns are closely related to AI ethics.** "Doing" AI ethics involves things already being done in many organizations developing ML. Well-established concerns related to AI ethics include cybersecurity and model robustness and safety. These concerns are a part of AI ethics but are nonetheless something most organizations developing ML systems are familiar with. Thus, it is possible to "unintentionally" tackle AI ethics to some extent, and some aspects of AI ethics may be closer to existing practices. This is also relevant for monitoring.

**MLOps may support the implementation of AI ethics** through sustainable MLOps [14]. Indeed, through an automated ML pipeline, MLOps may support the implementation of such principles as transparency (versioning, established process ... etc.). However, the practicalities of monitoring ethical attributes warrant consideration in MLOps contexts. For example: which (ethics-related) metrics are to be monitored constantly in an automated fashion? Which metrics should be checked only in certain situations, such as upon training or deploying a new model? Which metrics should result in automated alerts and what are the thresholds for such alerts for these metrics, and which metrics are less crucial and can be monitored more leisurely? We consider these interesting avenues for future research.

**AI ethics requires a framework.** To "do" AI ethics, one needs to define what it is in the given context [6, 46]. The most straightforward way of doing this is by utilizing an existing framework, as we have done in this paper (through AI HLEG [15]). This makes it possible for you define what it actually is you wish to monitor as well (e.g., which principles). Various guidelines [4, 5], tools [27], and methods (e.g. [7]) may be utilized for this purpose.

**Challenges with knowhow, communication, and collaboration.** ML development in

and of itself involves collaboration challenges as the rest of the development team needs to collaborate with the ML experts, which can be challenging in practice [24]. Similar collaboration challenges between different types of experts have been studied in the context of DevOps [47]. In terms of monitoring, Kourouklidis et al. [48] remark that *“domain experts in the area of ML, who produce the ML models, commonly lack the required expertise in the area of software engineering, needed to implement a robust and scalable monitoring solution”*, pointing to an issue improved collaboration could alleviate. Adding AI ethics expertise into this mix is arguably likely to only pose further challenges.

On this note, **real-world development contexts may also pose challenges for monitoring**. ML development capabilities are not something found in every company. In the case of commissioned systems developed by an external organization, the monitoring of ethics-related metrics is something that needs to be discussed between the involved parties.

## 6. Conclusions

In this paper, we have discussed AI ethics from the point of view of monitoring in general, as well as MLOps more specifically. We have presented a holistic typology of metrics for ML system monitoring, which we have then utilized to look at ML system monitoring overall, as well as in relation to AI ethics. This typology includes seven categories of metrics: (1) data metrics, (2) ML model metrics, (3) system/infrastructure metrics, (4) process metrics, (5) business metrics, (6) user metrics, and (7) domain-specific metrics. We argue that this typology, which is based on a breadth of existing literature from the fields of both software engineering and machine learning, provides one way of conceptualizing ML system monitoring in a holistic manner. Additionally, as a framework for the discussion on monitoring related to AI ethics, we have utilized the Ethics Guidelines for Trustworthy AI [15] (AI HLEG). Through this discussion, we have provided some examples of how AI ethics may manifest in ML system monitoring.

While this paper begins to illustrate the relevance of AI ethics for ML system monitoring, its main purpose is to encourage further discussion and studies into the topic. As one tangible future research suggestion, we recommend a literature review of what is currently known about ethics-related metrics in AI development. A large number of papers discussing ethics-related metrics already exists but this discussion is split across disciplines and the concepts utilized in these papers are hardly uniform.

## Acknowledgments

This work was partly funded by local authorities (“Business Finland”) under grant agreement ITEA-2020-20219-IML4E of the ITEA4 programme.

## Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

## References

- [1] B. Mittelstadt, Principles alone cannot guarantee ethical ai, *Nature Machine Intelligence* (2019) 1–7.
- [2] V. Vakkuri, K. Kemell, J. Kultanen, P. Abrahamsson, The current state of industrial practice in artificial intelligence ethics, *IEEE Software* 37 (2020) 50–57.
- [3] V. Vakkuri, K. Kemell, J. Kultanen, M. T. Siponen, P. Abrahamsson, Ethically aligned design of autonomous systems: Industry viewpoint and an empirical study, *EJBO - Electronic Journal of Business Ethics and Organization Studies* 27 (2022).
- [4] A. Jobin, M. Ienca, E. Vayena, The global landscape of ai ethics guidelines, *Nature Machine Intelligence* 1 (2019) 389–399.
- [5] T. Hagendorff, The ethics of ai ethics: An evaluation of guidelines, *Minds and Machines* (2020) 1–22.
- [6] E. Halme, M. Jantunen, V. Vakkuri, K.-K. Kemell, P. Abrahamsson, Making ethics practical: User stories as a way of implementing ethical consideration in software engineering, *Information and Software Technology* 167 (2024) 107379.
- [7] V. Vakkuri, K.-K. Kemell, M. Jantunen, E. Halme, P. Abrahamsson, ECCOLA – a method for implementing ethically aligned ai systems, *Journal of Systems and Software* 182 (2021) 111067. doi:<https://doi.org/10.1016/j.jss.2021.111067>.
- [8] K. K. Kemell, V. Vakkuri, F. Sohrab, How do ai ethics principles work? from process to product point of view, in: *Conference on Technology Ethics, Tethics, CEUR-WS, 2023*, pp. 24–38.
- [9] B. Fitzgerald, K. Stol, Continuous software engineering: A roadmap and agenda, *Journal of Systems and Software* 123 (2017) 176–189.
- [10] P. Rodríguez, A. Haghghatkhah, L. E. Lwakatare, S. Teppola, T. Suomalainen, J. Eskeli, T. Karvonen, P. Kuvaja, J. M. Verner, M. Oivo, Continuous deployment of software intensive products and services: A systematic mapping study, *Journal of Systems and Software* 123 (2017) 263–291. doi:<https://doi.org/10.1016/j.jss.2015.12.015>.
- [11] M. M. John, H. H. Olsson, J. Bosch, Towards mlops: A framework and maturity model, in: *2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2021*, pp. 1–8. doi:10.1109/SEAA53835.2021.00050.
- [12] A. Ghosh, A. Shanbhag, C. Wilson, Faircanary: Rapid continuous explainable fairness, in: *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society, 2022*, pp. 307–316.
- [13] Q. Lu, L. Zhu, X. Xu, J. Whittle, D. Douglas, C. Sanderson, Software engineering for responsible ai: An empirical study and operationalised patterns, in: *Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice, 2022*, pp. 241–242.
- [14] D. A. Tamburri, Sustainable mlops: Trends and challenges, in: *2020 22nd international symposium on symbolic and numeric algorithms for scientific computing (SYNASC), IEEE, 2020*, pp. 17–23.
- [15] Ethics Guidelines for Trustworthy AI, *Ethics guidelines for trustworthy ai*, 2019. URL: <https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai>.
- [16] L. Gao, M. Lu, L. Li, C. Pan, A survey of software runtime monitoring, in: *2017 8th IEEE*



- International Conference on Software Engineering and Service Science (ICSESS), 2017, pp. 308–313. doi:10.1109/ICSESS.2017.8342921.
- [17] T. Schröder, M. Schulz, Monitoring machine learning models: a categorization of challenges and methods, *Data Science and Management* 5 (2022) 105–116. doi:<https://doi.org/10.1016/j.dsm.2022.07.004>.
- [18] B. Kitchenham, J. Walker, A quantitative approach to monitoring software development, *Software Engineering Journal* 4 (1989).
- [19] S. Moreschini, F. Lomio, D. Hästbacka, D. Taibi, Mlops for evolvable ai intensive software systems, in: 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2022, pp. 1293–1294. doi:10.1109/SANER53432.2022.00155.
- [20] E. Ries, *The Lean Startup: How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, New York: Crown Business, 2011.
- [21] P. Kourouklidis, D. Kolovos, N. Matragkas, J. Noppen, Towards a low-code solution for monitoring machine learning model performance, in: Proceedings of the 23rd ACM/IEEE international conference on model driven engineering languages and systems: companion proceedings, 2020, pp. 1–8.
- [22] S. Mäkinen, H. Skogström, E. Laaksonen, T. Mikkonen, Who needs mlops: What data scientists seek to accomplish and how can mlops help?, in: 2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN), IEEE, 2021, pp. 109–112.
- [23] M. Naser, A. H. Alavi, Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences, *Architecture, Structures and Construction* (2021) 1–19.
- [24] G. Giray, A software engineering perspective on engineering machine learning systems: State of the art and challenges, *Journal of Systems and Software* 180 (2021) 111031.
- [25] S. Martínez-Fernández, J. Bogner, X. Franch, M. Oriol, J. Siebert, A. Trendowicz, A. M. Vollmer, S. Wagner, Software engineering for ai-based systems: A survey, *ACM Trans. Softw. Eng. Methodol.* 31 (2022). doi:10.1145/3487043.
- [26] J. Borenstein, F. S. Grodzinsky, A. Howard, K. W. Miller, M. J. Wolf, Ai ethics: A long history and a recent burst of attention, *Computer* 54 (2021) 96–102.
- [27] J. Morley, L. Floridi, L. Kinsey, A. Elhalal, From what to how: An initial review of publicly available ai ethics tools, methods and research to translate principles into practices, *Science and Engineering Ethics* 26 (2020) 2141–2168.
- [28] M. Sloane, J. Zakrzewski, German ai start-ups and “ai ethics”: Using a social practice lens for assessing and implementing socio-technical innovation, in: 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT ’22, Association for Computing Machinery, New York, NY, USA, 2022, p. 935–947.
- [29] V. Vakkuri, K.-K. Kemell, M. Jantunen, P. Abrahamsson, “this is just a prototype”: How ethics are ignored in software startup-like environments, in: V. Stray, R. Hoda, M. Paasi-vaara, P. Kruchten (Eds.), *Agile Processes in Software Engineering and Extreme Programming*, Springer International Publishing, Cham, 2020, pp. 195–210.
- [30] J. A. Siqueira De Cerqueira, A. Pinheiro De Azevedo, H. Acco Tives, E. Dias Canedo, Guide for artificial intelligence ethical requirements elicitation-re4ai ethical guide, in: 55th Hawaii International Conference on System Sciences, 2022.
- [31] N. Forsgren, M. Kersten, Devops metrics, *Commun. ACM* 61 (2018) 44–48. URL: <https://doi.org/10.1145/3211111>.

[//doi.org/10.1145/3159169](https://doi.org/10.1145/3159169). doi:10.1145/3159169.

- [32] R. Amaro, R. Pereira, M. M. da Silva, Capabilities and metrics in devops: A design science study, *Information & Management* 60 (2023) 103809. doi:<https://doi.org/10.1016/j.im.2023.103809>.
- [33] K.-K. Kemell, X. Wang, A. Nguyen-Duc, J. Grendus, T. Tuunanen, P. Abrahamsson, Startup metrics that tech entrepreneurs need to know, in: *Fundamentals of Software Startups: Essential Engineering and Business Aspects*, Springer, 2020, pp. 111–127.
- [34] K. Rodden, H. Hutchinson, X. Fu, Measuring the user experience on a large scale: user-centered metrics for web applications, in: *Proceedings of the SIGCHI conference on human factors in computing systems*, 2010, pp. 2395–2398.
- [35] T. Honglei, S. Wei, Z. Yanan, The research on software metrics and software complexity metrics, in: *2009 International Forum on Computer Science-Technology and Applications*, volume 1, 2009, pp. 131–136. doi:10.1109/IFCSTA.2009.39.
- [36] T. Mikkonen, J. K. Nurminen, M. Raatikainen, I. Fronza, N. Mäkitalo, T. Männistö, Is machine learning software just software: A maintainability view, in: D. Winkler, S. Biffl, D. Mendez, M. Wimmer, J. Bergsmann (Eds.), *Software Quality: Future Perspectives on Software Engineering Quality*, Springer International Publishing, Cham, 2021, pp. 94–105.
- [37] N. E. Fenton, M. Neil, Software metrics: roadmap, in: *Proceedings of the Conference on the Future of Software Engineering*, 2000, pp. 357–370.
- [38] S. Ackerman, O. Raz, M. Zalmanovici, A. Zlotnick, Automatically detecting data drift in machine learning classifiers, *arXiv preprint arXiv:2111.05672* (2021).
- [39] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, G. Zhang, Learning under concept drift: A review, *IEEE transactions on knowledge and data engineering* 31 (2018) 2346–2363.
- [40] M. Bano, D. Zowghi, A systematic review on the relationship between user involvement and system success, *Information and software technology* 58 (2015) 148–169.
- [41] H. H. Olsson, J. Bosch, From opinions to data-driven software r&d: A multi-case study on how to close the 'open loop' problem, in: *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*, 2014, pp. 9–16. doi:10.1109/SEAA.2014.75.
- [42] Y. Arifin, T. G. Sastria, E. Barlian, User experience metric for augmented reality application: A review, *Procedia Computer Science* 135 (2018) 648–656. URL: <https://www.sciencedirect.com/science/article/pii/S187705091831514X>. doi:<https://doi.org/10.1016/j.procs.2018.08.221>, the 3rd International Conference on Computer Science and Computational Intelligence (ICCSI 2018) : Empowering Smart Technology in Digital Era for a Better Life.
- [43] Z. Wang, N. Saxena, T. Yu, S. Karki, T. Zetty, I. Haque, S. Zhou, D. Kc, I. Stockwell, X. Wang, et al., Preventing discriminatory decision-making in evolving data streams, in: *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, 2023, pp. 149–159.
- [44] R. R. Hoffman, S. T. Mueller, G. Klein, J. Litman, Metrics for explainable ai: Challenges and prospects, *arXiv preprint arXiv:1812.04608* (2018).
- [45] TechNewsWorld, More than one in three firms burned by ai bias, 2022. URL: <https://www.technewsworld.com/story/more-than-one-in-three-firms-burned-by-ai-bias-87387.html>.
- [46] K.-K. Kemell, V. Vakkuri, What is the cost of ai ethics? initial conceptual framework and empirical insights, in: *International Conference on Software Business*, Springer, 2023, pp.

247–262.

- [47] M. S. Khan, A. W. Khan, F. Khan, M. A. Khan, T. K. Whangbo, Critical challenges to adopt devops culture in software organizations: A systematic review, *IEEE Access* 10 (2022) 14339–14349. doi:10.1109/ACCESS.2022.3145970.
- [48] P. Kourouklidis, D. Kolovos, J. Noppen, N. Matragkas, A model-driven engineering approach for monitoring machine learning models, in: *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, IEEE, 2021, pp. 160–164.