

Actionable Meta-Requirements for AI Robustness Auditing

JOSHUA OWOTOGBE, Jheronimus Academy of Data Science, Netherlands and Tilburg University, Netherlands

ROYA NASIRI, Jheronimus Academy of Data Science, Netherlands and Tilburg University, Netherlands

INDIKA KUMARA, Jheronimus Academy of Data Science, Netherlands and Tilburg University, Netherlands

DAMIAN ANDREW TAMBURRI, Jheronimus Academy of Data Science, Netherlands and University of Sannio, Italy

WILLEM-JAN VAN DEN HEUVEL, Jheronimus Academy of Data Science, Netherlands and Tilburg University, Netherlands

AI-enabled systems must remain dependable when the conditions under which they operate change (e.g., due to input perturbations, distribution shifts, or changes in external dependencies). This makes robustness auditing of such systems important for judging whether the available evidence is sufficient to support a defined robustness conclusion. However, designing an AI robustness-auditing framework is challenging. A key unresolved issue is determining the requirements such a framework should support; these requirements have not yet been consolidated. We address this gap through meta-requirements: higher-order requirements specifying what a class of auditing frameworks should support. In design science research, literature and domain experts are common sources of meta-requirements; accordingly, we derive the initial set from literature and validate and refine it with practitioners. Our systematic literature review retained 74 of 2,890 records and derived 43 meta-requirements across 16 categories. We evaluated them with 128 practitioners, obtaining 5,504 ratings and 586 substantive comments. Overall, 81.20% of ratings indicated agreement or strong agreement. In particular, 41 of the 43 literature-derived meta-requirements were classified as strongly supported. Practitioner feedback refined the requirements and identified an additional requirement concerning external components and supply-chain dependencies. The final set comprises 44 meta-requirements across 16 categories, of which 43 received quantitative validation. These meta-requirements can help organizations implement AI robustness-auditing frameworks, assess existing frameworks for completeness, and refine them.

CCS Concepts: • **Software and its engineering** → **Software verification and validation**; • **Computing methodologies** → **Artificial intelligence**; **Machine learning**.

Additional Key Words and Phrases: AI robustness, AI auditing, robustness auditing, meta-requirements, practitioner study, systematic literature review

1 Introduction

AI-enabled systems are increasingly deployed in high-stakes domains such as healthcare, transportation, criminal justice, and financial services, where incorrect predictions, inconsistent decisions, or failures under unexpected operating conditions can affect safety, individual rights, and access to essential services [27, 63, 68, 100]. In this study, we use *AI-enabled systems* to mean software systems in which at least one AI component contributes to the system’s functionality [75]. Such systems may also depend on non-AI components, including data pipelines, services, frameworks, APIs,

Authors’ Contact Information: Joshua Owotogbe, j.s.owotogbe@tilburguniversity.edu, Jheronimus Academy of Data Science, 's-Hertogenbosch, North Brabant, Netherlands and Tilburg University, Tilburg, North Brabant, Netherlands; Roya Nasiri, r.nasiri@tilburguniversity.edu, Jheronimus Academy of Data Science, 's-Hertogenbosch, North Brabant, Netherlands and Tilburg University, Tilburg, North Brabant, Netherlands; Indika Kumara, i.p.k.weerasinghadewage@tilburguniversity.edu, Jheronimus Academy of Data Science, 's-Hertogenbosch, North Brabant, Netherlands and Tilburg University, Tilburg, North Brabant, Netherlands; Damian Andrew Tamburri, datamburri@unisannio.it, Jheronimus Academy of Data Science, 's-Hertogenbosch, North Brabant, Netherlands and University of Sannio, Benevento, Benevento, Italy; Willem-Jan van den Heuvel, w.j.a.m.vdnHeuvel@tilburguniversity.edu, Jheronimus Academy of Data Science, 's-Hertogenbosch, North Brabant, Netherlands and Tilburg University, Tilburg, North Brabant, Netherlands.

infrastructure, and other software elements that can influence their overall behaviour. For concision, we use *AI auditing* throughout this paper to refer to the auditing of AI-enabled systems. In these domains, strong predictive performance under ordinary test conditions is insufficient. AI-enabled systems may produce different or degraded outputs when inputs are noisy or perturbed, data distributions shift, external dependencies change, or deployment conditions depart from those assumed during development [27, 30, 100, 117]. Robustness testing examines whether an AI-enabled system or its components maintain specified acceptable behaviour under defined changes to inputs or operating conditions. Such testing requires specifying the system or component being assessed, the expected behaviour, the disturbances or failure conditions considered, the evaluation procedure, and the acceptance criteria. [30, 46, 115, 117, 130]. Tests can produce observations about system behaviour under particular conditions, but an audit has a broader assurance role: it must judge whether the collected evidence is adequate for the stated robustness objective and make explicit the scope and limitations of that judgement.

This broader role makes the design of robustness auditing frameworks non-trivial. A useful framework must determine which concerns belong within the audit, what forms of evidence are relevant, how evaluation criteria should be established, how limitations should be documented, and when previously reached conclusions need to be revisited. These questions are increasingly consequential in regulatory and risk-management contexts. Article 15 of the European Union Artificial Intelligence Act (EU AI Act) requires high-risk AI systems to achieve an appropriate level of accuracy, robustness, and cybersecurity and to perform consistently in these respects [28, 89, 111]. Article 16 further requires providers to ensure compliance with these requirements; breaches of applicable provider obligations may be subject to administrative fines of up to EUR 15 million or, for undertakings, 3% of their total worldwide annual turnover for the preceding financial year, whichever is higher (Article 99(4)(a)) [28]. The National Institute of Standards and Technology Artificial Intelligence Risk Management Framework (NIST AI RMF), while voluntary, likewise addresses AI trustworthiness through characteristics including validity and reliability, safety, security, and resilience [112]. Such instruments set important expectations, but they do not, by themselves, provide an operational specification for a robustness auditing framework.

Relevant knowledge for such a specification spans several research streams. Technical robustness research develops benchmarks, metrics, perturbation methods, attacks, defences, and evaluation procedures for examining AI-enabled systems under adversarial attacks, data corruptions, data distribution shifts, and other changes in relevant inputs or operating conditions [1, 46, 69, 115, 117, 130]. Research on AI auditing, in turn, examines audit scoping, evidence collection and interpretation, governance, and documentation [79, 81, 99, 116]. However, to the best of our knowledge, no study has systematically reviewed such literature from the relevant domains and derived a catalog of requirements that can guide the development and evaluation of robustness auditing frameworks for AI-enabled systems (Section 3.3, Table 2). This study addresses this critical research gap.

Design science research (DSR) offers a complementary perspective by expressing general design needs for an artifact, independently of a particular implementation [47, 124, 132]. Following the DSR terminology, we use the term *meta-requirements* to refer to these implementation-independent requirements that capture what a family of auditing frameworks for AI-enabled systems is expected to accommodate. In DSR, the literature provides a basis for identifying such requirements, while practitioner knowledge provides evidence to assess and refine their practical relevance.

To derive an actionable set of meta-requirements for an AI robustness auditing framework, inspired by DSR methods, we conducted a mixed-method study [29, 97, 126] combining a systematic literature review with a practitioner validation survey. The database and supplementary searches identified 2,890 records, of which 74 met the eligibility criteria and quality-assessment threshold and were synthesised into 43 literature-derived meta-requirements across 16 categories and

three dimensions. We then evaluated these requirements through a survey of 128 practitioners using Likert ratings and open-text feedback to examine support, refinement, challenge, and additional requirements emerging from professional experience. The findings show substantial convergence between the literature and practitioner perspectives. Overall, 81.20% of the 5,504 ratings indicated agreement or strong agreement; 41 of the 43 literature-derived meta-requirements were classified as strongly supported, and the remaining two as supported. Practitioner feedback primarily refined wording, boundaries, access and evidence conditions, implementation assumptions, and lifecycle considerations, while also identifying an additional concern related to externally controlled dependencies. Integrating both evidence sources resulted in a final practitioner-informed set of 44 meta-requirements organised into 16 categories.

This article makes the following contributions:

- We derive a source-traceable set of AI robustness auditing meta-requirements from a systematic review of academic literature.
- We provide empirical evidence from practitioners on the perceived relevance and applicability of the literature-derived meta-requirements, using quantitative ratings and qualitative feedback to identify support, refinements, challenges, and previously underrepresented concerns.
- We discuss the implications of these findings for designing and evaluating future AI robustness auditing processes, methods, tools, and frameworks.
- We provide a replication package containing the review protocol, search strings, included studies, extraction forms, coding records, meta-requirement tables, survey instrument, and analysis outputs.

The remainder of this article is organised as follows. Section 2 introduces the key concepts and scope of the study. Section 3 reviews related work. Section 4 describes the research methodology, including the systematic literature review, meta-requirement derivation, and practitioner survey. Sections 5 and 6 present the resulting meta-requirements and practitioner validation findings, respectively. Section 7 discusses the findings and their implications. Section 8 addresses threats to validity, and Section 9 concludes the article.

2 Background

This section introduces the concepts needed to interpret the study. Table 1 summarises the terminology used throughout this article. The subsequent subsections show how robustness claims and evidence inform auditing and meta-requirements.

2.1 AI Robustness and Robustness Claims

A robustness assessment must specify what is being assessed (e.g., a model or AI-enabled system), the behaviour or outcome to be evaluated together with its acceptance criterion (e.g., prediction accuracy and the minimum accuracy that must be maintained), and the condition under which it is evaluated (e.g., adversarial inputs or distribution shift) [30, 117, 125]. These conditions matter because robustness is condition-specific: evidence obtained under one type of change does not establish robustness under every other type of change [30, 89]. The robustness requirement being assessed must therefore be expressed as a specific claim before its supporting evidence can be evaluated. For example, an image-classification robustness claim may require a classifier to maintain a specified minimum accuracy when test images are affected by Gaussian noise at a stated severity level; related tests may use blur, fog, or lighting changes to examine performance under other conditions [90, 125]. Although this example targets a model, the assessment object may extend to the wider AI-enabled system. While the model represents the trained AI component, an AI-enabled

Table 1. Key concepts and definitions used in this study.

Concept	Meaning in this study
AI-enabled system	A software system in which at least one AI component contributes to the system’s functionality [75].
AI audit	A systematic assessment of an AI-enabled system, its development or use, and relevant evidence against specified technical, organisational, legal, or ethical criteria [79, 81].
AI robustness	The ability of an AI-enabled system to maintain specified acceptable behaviour when relevant inputs or operating conditions change [30, 117, 125].
Robustness testing	Testing whether an AI-enabled system or its components maintain specified acceptable behaviour under defined changes to inputs or operating conditions. [8, 41, 41, 108, 125, 131]
Adversarial robustness	A form of robustness concerned with adversarial attacks, that is, deliberate attempts to cause incorrect or otherwise undesired system behaviour, for example by manipulating model inputs [72, 89, 125].
Non-adversarial robustness	A form of robustness concerned with naturally occurring changes, such as noise, image blur, lighting changes, sensor degradation, or distribution shift, where the data encountered during evaluation or deployment differ from those represented during model development [30, 54, 89, 125].
Threat model	The assumptions about what an attacker can know and what actions the attacker can perform against an AI-enabled system or model [19].
Robustness metric	A quantitative measure used to report robustness, such as performance under specified test conditions, a robustness bound, or the probability that a robustness condition is satisfied [125].
Acceptance criterion	The condition used to determine whether an observed robustness result meets the required level of performance, such as a minimum accuracy, a maximum permitted output change, or a maximum probability of error [125].
Robustness claim	An explicit assertion that an identified system, model, or component will maintain stated acceptable behaviour under specified conditions and according to stated acceptance criteria [30, 110].
Robustness evidence	Information used to assess whether a robustness claim is supported, including test results, monitoring outputs, logs, analyses, and documentation [83, 99, 110].
AI robustness auditing	A structured assessment of whether available evidence supports a robustness claim within a defined assessment scope and against stated criteria [14, 99, 110].
Meta-requirement for AI robustness auditing	A requirement that states what robustness auditing approaches should support across different AI-enabled systems, while allowing the specific method used to vary with the system and audit context [47, 74, 124, 132].
Accuracy	The extent to which the outputs or predictions produced by an AI-enabled system are correct with respect to its intended task and the evaluation measure being used [89].
Computational cost	The amount of computational resources consumed or required to develop, evaluate, or operate an AI-enabled system, such as processing time, memory, energy, or computing capacity [19].
Explainability	The extent to which the outputs, decisions, or behaviour of an AI-enabled system can be understood or explained to people who need to interpret, assess, or act on them [72].
Fairness	The extent to which an AI-enabled system avoids unjustified differences in outcomes for affected people or groups based on their characteristics [72, 77, 85, 127].
Bias	A systematic tendency in data, models, or system behaviour that can produce consistently different or poorer outcomes for particular people, groups, or cases, and may contribute to unfairness [77].
Safety	The extent to which the behaviour or outputs of an AI-enabled system avoid causing or contributing to harm to people, property, or the environment [72].
Privacy	The extent to which an AI-enabled system protects personal or sensitive information from unauthorized access, disclosure, leakage, or inference [127].

system may also include services, frameworks, toolkits, platforms, infrastructure, and other software components [125, 131]. Accordingly, this study considers the robustness of the overall AI-enabled system, while allowing individual robustness claims to focus on a particular model or other system component.

2.2 AI Robustness Auditing

Once a robustness claim is specified, an AI robustness audit evaluates the available evidence against that claim, the assessment scope (i.e., the system or components included in the audit), and the criteria used to judge whether the claim is supported. The evidence may include test results, monitoring outputs, logs, data, documentation, and development or deployment records [14, 83, 99, 110]. Testing the robustness of an AI-enabled system or its components provides one type of audit evidence. For the image-classification claim above, the test could apply the specified noise to the test images and measure whether the classifier maintains the required accuracy [125]. Beyond such test results, the evidence available to the audit depends on the auditor’s level of access. With black-box access, the auditor can query the system and observe its outputs; with white-box access, the auditor may also inspect internal model information, such as model parameters (weights) and intermediate computations (activations); and with outside-the-box access, development and deployment information, such as code, documentation, data, and internal evaluation results, may also be available [14].

2.3 Meta-Requirements for Robustness Auditing

The preceding discussion shows why robustness auditing cannot rely on a single fixed procedure. The relevant claim, assessment object, available evidence, and level of auditor access can differ across AI-enabled systems, so a test or evaluation method suitable for one case may not be suitable for another [14, 30, 116, 117]. Despite these differences, robustness audits share common needs, including defining what is being assessed, obtaining and evaluating relevant evidence, and documenting the basis for conclusions [14, 99, 110]. These differences in how robustness audits are

conducted, together with the needs they share, motivate the use of meta-requirements to specify what robustness auditing approaches should address across different AI-enabled systems.

Design science research uses meta-requirements to express needs that should be addressed across a class of artefacts, methods, or approaches, while allowing different solutions to address those needs in different ways [38, 47, 66, 74, 124, 132]. In this study, a robustness auditing meta-requirement is *actionable* when it identifies a concern that can be examined in an audit and can guide the design or assessment of robustness auditing approaches, while the specific way in which that concern is addressed may vary with the AI-enabled system and audit context. Practitioner evaluation assessed the perceived relevance and practical applicability of these meta-requirements, but their effectiveness when applied in real AI robustness audits has not yet been evaluated.

3 Related Work

This section positions our study against prior work on challenges in robustness assessment, robustness practices across the AI lifecycle, and existing approaches to AI auditing.

3.1 Challenges Relevant to Auditing AI Robustness

Prior research highlights several challenges relevant to auditing AI robustness. First, robustness evaluations can differ in the attacks or input changes considered, their severity, the datasets used, the robustness metrics reported, and the acceptance criteria applied [30, 72, 125]. These differences make results difficult to compare and can limit the extent to which evidence from one evaluation setting supports conclusions in another [30, 51, 54]. Second, robustness failures can arise from interactions among multiple system layers and components rather than from the model alone [9, 131]. This makes it difficult to determine whether model-level testing is sufficient, which components should be included in the audit scope, and what evidence is needed for a system-level conclusion [14, 82, 99]. Third, thorough robustness assessment can require substantial computation, execution time, and infrastructure, particularly for verification, large-scale testing, and runtime monitoring [39, 125]. These demands can limit the extent of testing and, consequently, the strength of the resulting audit evidence. Fourth, robustness interventions can involve trade-offs with predictive accuracy, efficiency, explainability, fairness, privacy, usability, or deployment cost [16, 53, 85, 107, 109, 117, 130]. Such trade-offs can complicate audit conclusions because improvements in robustness may affect other system properties that are also relevant to acceptance. Finally, robustness evidence can become outdated after deployment. Changes in production data, the deployed model, or deployment conditions can make earlier assessment results less representative of the current system and may require monitoring or reassessment [8, 19, 70]. Together, these challenges motivate audit-oriented requirements addressing evaluation conditions, assessment scope, resource constraints, trade-offs, reassessment, and the basis and limitations of robustness conclusions.

3.2 Robustness Practices Across the AI Lifecycle

Prior research places robustness-related activities across the AI lifecycle, including data preparation, model training, robustness evaluation, deployment, and continuous monitoring [8, 69, 114]. First, at the data-preparation stage, relevant practices include checking and cleaning missing, inconsistent, or noisy data; augmenting the available data with additional examples; and constructing test data that represent relevant conditions, such as noise, blur, fog, lighting changes, or differences between development and deployment data [51, 90, 108, 114, 125]. Second, during model training, robustness can be improved through adversarial training, which exposes the model to adversarial examples; robust optimisation, which minimises worst-case loss under bounded input changes; regularisation techniques such as

dropout; and architecture modifications designed to improve robustness [1, 51, 80, 114, 129, 130]. Third, robustness evaluation examines how a trained model behaves under specified conditions. It can include adversarial testing, testing with common input changes, and evaluation on data from different distributions [51, 125]. Formal or statistical verification can additionally assess whether a specified robustness condition holds within a defined input region or with a stated probability [125]. These assessment activities examine whether a trained model satisfies specified robustness conditions and should therefore be distinguished from techniques intended to improve robustness [30]. Finally, after deployment, robustness-related activities include estimating prediction confidence or uncertainty, rejecting predictions when confidence is insufficient, detecting inputs that differ from the training distribution or have been deliberately manipulated, and monitoring model behaviour during operation [39, 80]. Production processes can also monitor distribution or concept drift, retain relevant documentation and logs, and define when model updates or retraining are required [8, 70]. Across the AI-enabled system lifecycle, robustness auditing must relate evidence to the assessed robustness claim, system or component within scope, acceptance criteria, and assessment conditions [82, 99, 116]. It must also document evidence limitations and determine when changes require reassessment. These needs motivate the consolidation of technical and procedural knowledge into audit-oriented meta-requirements.

3.3 Limitations of Prior Work on AI Robustness Auditing

Prior work relevant to AI robustness auditing contributes complementary foundations from different research streams. Robustness-focused studies clarify how robustness can be understood and evaluated. Tocchetti et al. [117] systematise robustness terminology and organise prior work around robustness-improvement, application-specific, and evaluation approaches, while Freiesleben and Grote [30] develop a conceptual account of robustness and distinguish robustness testing within an audit from techniques intended to improve robustness. Together, these studies provide a basis for robustness assessment, but they do not define the requirements of a complete robustness auditing process. Studies on AI auditing provide guidance on how audits are organised and carried out. Raji et al. [99] propose a five-stage internal audit process that moves from defining the audit scope and examining the system context to collecting evidence, testing the system, and reflecting on the findings. Mökander et al. [82] organise LLM auditing across the organisation developing the model, the model itself, and its downstream applications. Simbeck [110] structures auditing from defining the scope and assessment criteria through evaluation to subsequent monitoring. Together, these studies show how AI audits can be organised, but they do not define robustness-specific audit requirements.

Regulatory and trustworthy-AI studies identify requirements and obligations that AI-enabled systems are expected to satisfy. Nolte et al. [89] analyse the robustness and cybersecurity requirements of the EU AI Act and their implementation implications. Thelisson and Verma [116] examine how high-risk AI systems can be assessed against regulatory requirements before market entry and propose supporting assessment tools. Díaz-Rodríguez et al. [26] analyse seven requirements for trustworthy AI, including technical robustness and safety. Together, these studies clarify what AI-enabled systems should satisfy, but not the meta-requirements needed to design and conduct robustness audits. Requirements-oriented studies provide the closest methodological precedent. Mäntymäki et al. [74] derive seven meta-requirements for organisational AI governance through design science research, combining literature-based premises with insights from an industry-academia project involving organisational design partners. Zacharias et al. [132] derive three meta-requirements for explainable feature selection and evaluate a resulting software artefact with practitioners and researchers. Von Zahn et al. [123] derive four meta-requirements for assessing AI conformity, with their implemented assessment centred on fairness, explainability, and performance. Boufaied et al. [10] provide practitioner evidence on how fairness requirements are interpreted and managed across the AI/ML lifecycle.

These studies provide useful methods for deriving and evaluating requirements, but the reviewed work does not consolidate robustness-specific audit requirements for stating what robustness is being claimed, which parts of the AI-enabled system are covered, which failures or operating changes should be tested, what evidence is needed to support the audit conclusion, and when reassessment is required. We also did not identify a study that derives such meta-requirements from the literature and evaluates them with practitioners (Table 2).

Table 2. Comparison of representative studies relevant to AI robustness auditing meta-requirements.

Ref.	Year	Objective	Method	Robustness Focus	Audit/Assur.	Req./MR Output	Pract. Eval.	Findings / Limitation
[99]	2020	Develop an end-to-end internal framework for algorithmic auditing for development of AI systems.	Framework synthesis informed by auditing practices in adjacent disciplines; illustrative cases	●	✓	-	✗	Provides a broad internal AI auditing framework with staged evidence collection and testing, but no robustness-specific meta-requirements.
[132]	2022	Develop a design framework for explainable feature selection and instantiate it as a software artefact.	Design Science Research; literature and legal-document review; two expert focus groups	✗	✗	3 meta-requirements + design principles	✓ (n = 12)	Derives and evaluates meta-requirements for explainable feature selection, not robustness auditing.
[74]	2023	Derive meta-requirements for organisational AI governance frameworks.	Design Science Research; literature-derived premises; industry-academia collaboration with organisational design partners	✗	●	7 meta-requirements	✓	Derives meta-requirements for AI governance with organisational design-partner input, but does not develop robustness-specific auditing meta-requirements.
[30]	2023	Develop a unifying conceptual account of ML robustness based on targets and modifiers.	Conceptual analysis	✓	●	-	✗	Provides a conceptual robustness framework and distinguishes robustness testing from robustness improvement, but no robustness auditing requirements framework.
[26]	2023	Connect trustworthy-AI principles, ethics, regulation, and technical requirements.	Multidisciplinary conceptual analysis	●	●	7 trustworthy-AI requirements analysed	✗	Analyses seven trustworthy-AI requirements, including robustness and safety, but does not derive robustness auditing meta-requirements.
[82]	2023	Develop a three-layered blueprint for auditing large language models.	Literature review; conceptual synthesis	●	✓	7 design claims; no meta-requirements	✗	Proposes coordinated governance, model, and application audits and seven design claims, but no robustness-specific meta-requirements.
[110]	2024	Develop an auditing framework for learning analytics systems covering fairness, transparency, and robustness.	Conceptual synthesis of audit approaches and domain-specific criteria	✓	✓	-	✗	Provides a four-phase audit process in which robustness is assessed alongside fairness and transparency, but no robustness-specific meta-requirements.
[117]	2025	Reconcile terminology concerning AI robustness and organise robustness-improvement and evaluation approaches from a human-centred perspective.	Systematic survey; thematic analysis	✓	✗	-	✗	Systematises robustness concepts, methods, applications, and evaluation approaches, but does not develop an framework of audit requirements.
[89]	2025	Analyse robustness and cybersecurity requirements in the EU AI Act and their implementation implications.	Legal analysis of AI Act provisions; comparison with ML literature	✓	●	Legal requirements analysed	✗	Clarifies legal robustness and cybersecurity requirements and implementation challenges, but does not develop robustness auditing meta-requirements.
[123]	2025	Develop and evaluate a design framework and software artefact for AI conformity assessment.	Design Science Research; literature and legal-document review; seven expert interviews; five focus groups	✗	✓	4 meta-requirements + design principles	✓ (n = 21)	Provides four practitioner-evaluated meta-requirements for AI conformity assessment, with the implemented assessment centred on fairness and explainability rather than robustness.
[10]	2026	Investigate how practitioners interpret and manage fairness requirements across the AU/ML software-development lifecycle.	Qualitative interview study; 26 semi-structured interviews; thematic analysis	✗	✗	Fairness-requirement findings + challenges framework	✗	Provides practitioner evidence on fairness requirements, but does not derive or evaluate robustness auditing meta-requirements.
This study	2026	Derive literature-grounded meta-requirements for AI robustness auditing and evaluate them with practitioners.	Systematic literature review + practitioner survey	✓	✓	43 literature-derived + 1 practitioner-informed meta-requirement	✓ (n = 128)	Derives 43 literature-based meta-requirements, evaluates them with practitioners, and incorporates one additional meta-requirement from qualitative feedback, resulting in a final set of 44 meta-requirements.

Note. ✓ = explicitly addressed; ● = partially addressed; ✗ = not addressed; - = no requirements or meta-requirements reported. *Robustness Focus* indicates whether robustness is a central concern of the study. *Audit/Assur.* indicates explicit audit, assurance, or regulatory-assessment coverage. *Req./MR Output* distinguishes analysed requirements from explicitly derived meta-requirements and design principles. *Pract. Eval.* indicates empirical evaluation of a proposed framework, requirements, or artefact with practitioners or domain experts.

4 Methodology

This study adopts a mixed-methods design [29, 97, 126] to derive and validate meta-requirements for AI robustness auditing. It combines a systematic literature review, used to derive the literature-grounded meta-requirements, with a practitioner validation survey, used to assess and refine them based on professional experience. The review component was guided by established systematic review guidelines [33, 34, 56]. The research process comprises four phases: (1) literature search and screening, (2) meta-requirement extraction and refinement, (3) practitioner validation survey, and (4) meta-requirement synthesis and reporting (see Figure 1).

4.1 Research Questions

This study is guided by two research questions. RQ1 identifies the meta-requirements for AI robustness auditing, while RQ2 examines practitioners’ perspectives on those meta-requirements (see Table 3).

RQ1 is addressed through Phases 1 and 2, which identify, code, and consolidate audit-relevant evidence from the literature into meta-requirements. RQ2 is addressed through Phases 3 and 4, which analyse practitioner ratings and open-text comments and integrate this validation evidence into the final set of meta-requirements.

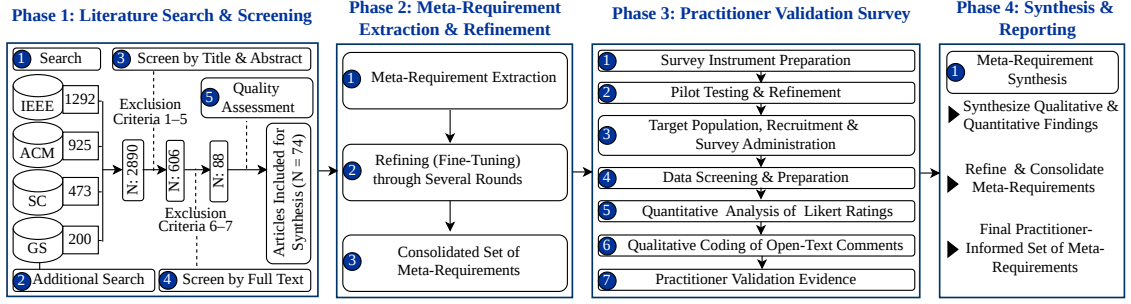


Fig. 1. Overview of the four-phase research method used to derive, refine, validate, and synthesize the meta-requirements.

Table 3. Research questions and motivation.

Research Question	Motivation
RQ1: What meta-requirements should an AI robustness auditing framework satisfy?	This question identifies the requirements that should guide the design and conduct of AI robustness auditing.
RQ2: What are practitioners' perspectives on the proposed meta-requirements for AI robustness auditing?	This question examines practitioners' support for, qualifications of, challenges to, and suggested refinements or additions to the proposed meta-requirements.

4.2 Phase 1: Literature Search and Screening

Following established systematic-review guidance [56], we constructed the search around AI/ML systems, robustness-related concerns, and audit-relevant assessment activities. To focus on studies with practical relevance to robustness auditing, the search strings additionally required terms referring to practitioners (practitioner, developer, engineer, or stakeholder) and terms referring to empirical studies (interview or survey). Pilot searches indicated that removing these restrictions substantially increased the number of irrelevant records. We therefore retained the restrictions to improve search precision, while recognising that they could exclude relevant studies that did not use this terminology. Candidate studies were retrieved from three academic databases: IEEE Xplore ($N = 1,292$), ACM Digital Library ($N = 925$), and Scopus ($N = 473$) (see Phase 1 in Figure 1). We selected these databases because they provide broad coverage of software engineering, artificial intelligence, and information systems research [18]. The search was restricted to studies published between January 2019 and February 2026, and all results were exported in BibTeX and CSV formats for reference management and data processing. Table 4 reports the final search strings and per-database retrieval counts.

Table 4. Search Strings and Results

Digital Library	Search String	Retrieved Numbers	Articles
ACM	[[Abstract: "robustness" OR [Abstract: resilient] OR [Abstract: reliab] OR [Abstract: dependab]] AND [[Abstract: "artificial intelligence" OR [Abstract: "machine learning" OR [Abstract: "algorithmic" OR [Abstract: "ai" OR [Abstract: "ml"]]] AND [[Abstract: audit] OR [Abstract: assess] OR [Abstract: test] OR [Abstract: evaluat]]] AND [[Abstract: practitioner] OR [Abstract: developer] OR [Abstract: engineer] OR [Abstract: stakeholder]]] AND [[Abstract: interview] OR [Abstract: survey]]]	925	
Scopus	TITLE-ABS-KEY (("robustness" OR "resilien" OR "reliab" OR "dependab") AND ("artificial intelligence" OR "machine learning" OR "algorithmic" OR "AI" OR "ML") AND ("audit" OR "assess" OR "test" OR "evaluat") AND ("practitioner" OR "developer" OR "engineer" OR "stakeholder") AND ("interview" OR "survey"))	473	
IEEE Xplore	("All Metadata":robustness OR "All Metadata":resilien OR "All Metadata":reliab OR "All Metadata":dependab) AND ("All Metadata":artificial intelligence OR "All Metadata":machine learning OR "All Metadata":algorithmic OR "All Metadata":AI OR "All Metadata":ML) AND ("All Metadata":audit OR "All Metadata":assess OR "All Metadata":test OR "All Metadata":evaluat) AND ("All Metadata":practitioner OR "All Metadata":developer OR "All Metadata":engineer OR "All Metadata":stakeholder) AND ("All Metadata":interview OR "All Metadata":survey)	1292	
Total	-	2690	

To complement the database searches, we conducted a supplementary Google Scholar search, following prior systematic reviews that used it to extend database-based retrieval [3, 31, 49]. We examined the first 20 pages using terms reflecting the same AI/ML, robustness, audit, practitioner, and empirical-study scope as the database searches, and

applied the same eligibility criteria. The database searches yielded 2,690 records, and the supplementary Google Scholar search added 200, resulting in 2,890 records for title/abstract screening. Following established SLR guidance [12, 35, 50], Table 5 presents the inclusion and exclusion criteria.

Table 5. Inclusion and exclusion criteria.

Inclusion Criteria	Exclusion Criteria
IC1: Studies must be written in English.	EC1: Studies not written in English.
IC2: Studies must be published between January 2019 and February 2026.	EC2: Studies published outside the selected time window.
IC3: Studies must be available in full text.	EC3: Studies whose full text is unavailable or inaccessible.
IC4: Studies must be academic research publications, including journal articles, conference papers, or workshop papers of at least six pages.	EC4: Duplicate records, papers shorter than six pages, books, editorials, abstracts, posters, proceedings summaries, presentation-only papers, or other publication types lacking sufficient research content.
IC5: Studies must examine AI, machine learning, deep learning, large language models, algorithmic systems, or AI-enabled software systems as the object of study.	EC5: Studies outside the AI/ML domain, or studies that use AI/ML only as an analytical tool for prediction, classification, diagnosis, forecasting, optimisation, sensing, or domain analysis without examining the AI/ML system itself.
IC6: Studies must address robustness or related dependability concerns of AI/ML systems, such as reliability, resilience, safety, fault tolerance, failure behaviour, distribution shift, out-of-distribution behaviour, adversarial robustness, or failure analysis.	EC6: Studies that use robustness, reliability, or resilience only in a non-AI sense, or only to describe general predictive performance, efficiency, usability, adoption, trust, sustainability, educational outcomes, physical-system performance, or organisational resilience.
IC7: Studies must connect AI/ML robustness or dependability to audit-relevant activities, including assessment, testing, evaluation, verification, validation, monitoring, assurance, compliance, governance, documentation, or evidence generation.	EC7: Studies focused only on adjacent topics such as fairness, privacy, explainability, cybersecurity, traceability, forensics, ethics, policy, adoption, education, usability, or general trust without a direct connection to AI/ML robustness or dependability assessment.

The selection process proceeded in five steps (see Phase 1 in Figure 1). Two researchers independently screened all 2,890 records at title/abstract level using IC1–IC5 and EC1–EC5, achieving 96.23% agreement (Cohen’s $\kappa = 0.882$). The 109 disagreements were resolved by consensus, leaving 606 studies for full-text review. The same researchers independently assessed the 606 full texts using IC6–IC7 and EC6–EC7, achieving 94.22% agreement ($\kappa = 0.741$); we resolved 35 disagreements by consensus. After reconciliation, 88 studies were retained for quality assessment.

4.2.1 Quality Assessment. The 88 eligible studies were assessed using six quality criteria adapted from prior SLR practices [6, 11]. The criteria examined whether: (QA1) the study objective or research question was clearly stated; (QA2) the study design or methodology was suitable for that objective; (QA3) the AI/ML system context was sufficiently described; (QA4) the data or evidence sources and analysis procedures were sufficiently described; (QA5) the findings and conclusions were supported by the reported evidence and analysis; and (QA6) relevant assumptions, limitations, or threats to validity were reported. Each criterion was scored as *Yes* (1), *Partly* (0.5), or *No* (0). Studies scoring at least 4 out of 6 were retained for the final synthesis. The first researcher scored all 88 studies against QA1–QA6, and the second researcher reviewed the completed quality assessments for all studies. Any differences identified during this review were discussed and resolved by consensus. After quality assessment, 74 studies were retained. The QA1–QA6 scores, total quality score, and final inclusion decision for each of the 88 assessed studies are provided in the replication package (see the Data Availability statement).

Among the 74 studies retained after quality assessment, publication activity was limited in 2020 ($N = 4$) and 2021 ($N = 2$), followed by 11 studies in 2022, 14 in 2023, 13 in 2024, and a peak of 22 in 2025 (Figure 2). The 2026 count ($N = 8$) reflects the February 2026 search cut-off. Journal articles accounted for 50 studies, followed by conference papers ($N = 23$) and one workshop paper. Figure 3 shows that 33 studies appeared in Q1 venues and 28 in unranked venues, followed by Q2 ($N = 6$), A* ($N = 5$), and A ($N = 2$) venues. Venue ranking was reported descriptively and was not used as a proxy for study quality or as an exclusion criterion. Review rigour was instead supported through predefined eligibility criteria, independent screening, consensus resolution, and traceable evidence extraction. Following Galster et al. [32], lower-ranked and unranked studies were retained when they met the eligibility criteria and contributed relevant evidence.

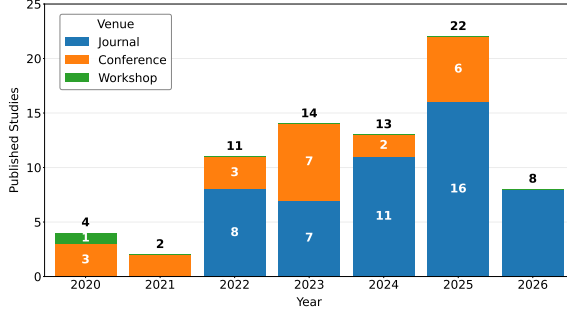


Fig. 2. Annual distribution of the reviewed studies by publication venue type.

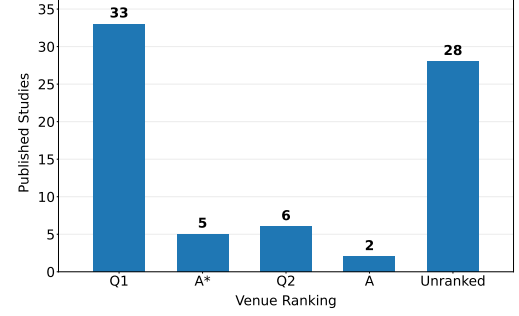


Fig. 3. Distribution of the reviewed studies by venue ranking.

4.3 Phase 2: Meta-Requirement Extraction and Refinement

The 74 retained studies were analysed using a structured qualitative coding procedure to derive meta-requirements for AI robustness auditing [17, 87]. The coding focused on how studies define robustness claims, specify audit scope, evaluate robustness, document evidence, assign audit responsibilities, involve stakeholders, monitor deployed systems, report limitations, and support reassessment. For each study, we used a standard extraction form to record the bibliographic details, AI/ML context, robustness concern, audit or evaluation activity, supporting quotation, extracted pattern, candidate category, and associated meta-requirement. The completed extraction records are provided in the replication package (see the Data Availability statement) and link each supporting quotation to its extracted pattern, candidate category, and literature-derived meta-requirement.

The first stage applied open coding [17] to identify audit-relevant passages and summarise each as an extracted pattern. In the second stage, axial coding grouped related patterns by underlying audit concern, including patterns that spanned different domains and system contexts. For example, evidence concerning the effects of robustness interventions on accuracy, fairness, and explainability was progressively grouped into the broader concern of trade-off measurement (see Figure 4).

The third stage converted the candidate categories into meta-requirements using explicit merge, separation, and retention criteria, following prior studies that consolidated lower-level evidence into higher-level tasks, categories, or requirements [29, 87, 97]. Candidate categories were merged when they represented the same auditing requirement despite differences in terminology, technique, or application context. They were kept separate when merging them would remove a different audit responsibility, evidence requirement, or audit decision. A candidate category was retained as a meta-requirement when it was within the study scope, supported by traceable source evidence, applicable beyond a single technique or system, and not already represented by another meta-requirement.

The resulting categories were assigned to three higher-level dimensions according to their primary purpose, following the use of explicit classification criteria in prior empirical studies [104, 119]. Categories were classified as *Human* when they primarily concerned people, roles, governance, competence; as *Functional* when they primarily concerned audit activities or capabilities; and as *Quality* when they primarily concerned properties of the auditing approach itself. Where a category related to more than one dimension, its dominant purpose determined the assignment. Recurrence across studies increased the supporting evidence for a concern but was not used as a fixed inclusion threshold. A

concern could therefore be retained when it met the criteria above even if it appeared in relatively few studies, while repeated evidence for the same higher-level concern did not result in additional meta-requirements.

Following established principles for writing quality requirements [44, 45], we refined the resulting meta-requirements for clarity, lack of ambiguity, atomicity, unique identification, linkage to supporting source evidence, audit assessability, and consistency across the set. We formulated each meta-requirement as an abstract, technology-agnostic statement specifying what a robustness auditing framework should support, require, or make explicit, and assigned each an identifier, category, and requirement statement. One researcher conducted the extraction, coding, and initial meta-requirement formulation. A second researcher independently reviewed the extracted patterns, category assignments, and resulting meta-requirements. The team resolved disagreements and ambiguities through discussion and consensus. This process produced 43 literature-derived meta-requirements across 16 categories, which formed the basis of the practitioner validation survey in Phase 3. Table 7 reports these together with MR-8.4, which was added after the survey and was therefore not rated by respondents. The replication package contains the corresponding extraction records, including the source passage, extracted pattern, category assignment, and resulting meta-requirement, allowing readers to inspect how the source evidence was consolidated.

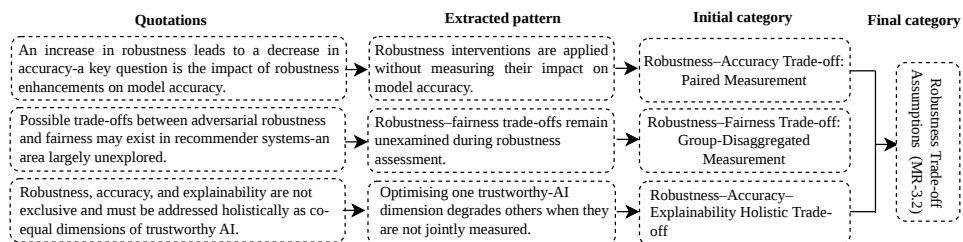


Fig. 4. Example of how extracted literature evidence is coded and consolidated into MR-3.2, Robustness Trade-off Assumptions.

4.4 Phase 3: Practitioner Validation Survey

The 43 meta-requirements derived in Phase 2 were evaluated through a practitioner survey addressing RQ2. Following established survey guidance in software engineering, we combined Likert-scale ratings with optional open-text feedback to assess practitioner support, interpretation, and suggested refinements, challenges, and additional concerns [57, 58, 71]. Comparable studies have similarly used practitioner surveys to assess and refine previously identified requirements and framework elements [121, 134].

4.4.1 Survey Instrument Preparation. We developed a Google Forms questionnaire comprising 44 numbered questions. First, the survey opened with a study-information and consent section describing the research purpose, expected completion time, voluntary participation, confidentiality arrangements, and researcher contact information. Second, Q2-Q5 collected demographic information, including age, gender, country or region, and education, while Q6-Q11 captured participants’ professional profile, including AI/ML experience, robustness-assessment or auditing experience, academic and professional experience levels, AI/ML-related role, and involvement in robustness-related activities.

Third, Q12-Q44 formed the validation section. The 43 meta-requirements were distributed across 16 matrix questions, one for each validation category, with each meta-requirement presented as a separate rating row. Respondents rated each meta-requirement using a five-point Likert scale from 1 = *strongly disagree* to 5 = *strongly agree*. Each matrix question

was followed by an optional open-text question for comments or suggested changes, and the final question invited feedback on missing requirements, duplication, unclear wording, implementation barriers, or other improvements. The complete questionnaire is provided in the replication package (see the Data Availability statement).

4.4.2 Pilot Testing and Refinement. Following prior practitioner-survey practice [121, 126], we piloted the survey from 24 June to 5 July 2026 with eight participants with relevant AI/ML, robustness, or auditing experience. The pilot assessed completion time, clarity of instructions and terminology, interpretation of the category descriptions and meta-requirements, suitability of the five-point scale, possible overlap, and usability across devices. Reported completion times ranged from approximately 15 to 30 minutes, with some reviewers finding completion more difficult on mobile devices. Based on the feedback, we shortened the introductory text, clarified category descriptions and meta-requirement labels, simplified wording, and retained the five-point scale and optional definitions. The eight pilot participants were not included in the main survey sample, and their responses were excluded from the main dataset and analysis.

4.4.3 Target Population, Recruitment, and Survey Administration. Eligible participants were researchers and practitioners worldwide with academic, professional, or applied experience in AI/ML systems and related activities, including robustness assessment, model evaluation, adversarial evaluation, AI auditing, AI safety, model reliability, or assurance. We used purposive non-probability sampling because the study required participants with relevant experience in areas related to AI robustness auditing, while the specialised target population could not be comprehensively enumerated for probability sampling [13, 29, 59, 126]. Potential participants were identified through GitHub and LinkedIn, channels that have also been used to recruit specialised practitioners in prior software engineering surveys [29, 121, 126]. On GitHub, we searched relevant repositories and contributors using robustness-related queries and GitHub’s *Best match* ranking. Representative queries included "AI robustness", "robustness auditing", "adversarial robustness", and "ML robustness". Approximately 1,200 potential participants were identified through GitHub and 520 through LinkedIn. Survey invitations were sent to all 1,720 identified candidates. We received 143 responses, corresponding to a response rate of 8.31%. After applying the predefined response-screening criteria, 15 responses were excluded, leaving 128 valid responses for analysis. Survey invitations briefly introduced the study, stated the eligibility criteria, and directed potential participants to the study-information and consent page of the questionnaire (see Section 4.4.1). The study received ethical clearance from the Research Ethics and Data Management Committee of the Tilburg School of Humanities and Digital Sciences, Tilburg University (REDC 2026.04). The main survey was administered through Google Forms from 8 July to 15 August 2026, and informed consent was obtained before participation. After completing the survey, respondents could optionally enter a draw for a €100 gift card [23]. Contact information submitted for the draw was collected separately from the survey responses.

4.4.4 Data Screening and Preparation. Following prior practitioner-survey research [126], we defined inclusion and exclusion criteria for screening the survey responses, as shown in Table 6. The inclusion criterion required respondents to demonstrate relevant academic, professional, or practical experience in at least one of the study’s target areas, as assessed from their professional-profile responses. Responses containing inconsistent, apparently random, or meaningless answers were considered for exclusion, as were responses without informed consent. We received 143 responses to the main survey. After applying the predefined screening criteria, 15 responses were excluded under E1, leaving 128 valid responses for quantitative analysis. The responses were exported from Google Forms and processed in Python by identifying the 43 meta-requirement rating columns, standardising response labels, converting the five-point Likert

responses to values from 1 to 5, checking missing values, assigning anonymous respondent identifiers, and transforming the ratings into long format. The resulting dataset contained 5,504 ratings, with no missing meta-requirement values. Open-text responses from the 128 valid respondents were screened separately because completion of these fields was optional. Across the 16 category-level comment fields and the final overall-comment field, 670 entries were extracted. Empty responses, uninterpretable statements, and comments without analytically usable feedback were removed, leaving 586 substantive comments for qualitative coding and synthesis. Each retained comment was linked to its anonymous respondent identifier, source category, and position within the corresponding response batch to preserve traceability.

Table 6. Inclusion and exclusion criteria for selecting valid responses.

Inclusion	I1:	The respondent must demonstrate relevant academic, professional, or practical experience in AI/ML systems, AI robustness assessment or auditing, robustness testing, model evaluation, adversarial evaluation, AI safety, model reliability, or related assurance activities.
Exclusion	E1: E2:	A response is excluded if it contains inconsistent, apparently random, or meaningless answers. A response without informed consent is excluded.

4.4.5 Quantitative Analysis of Likert Ratings. The 5,504 Likert ratings were analysed descriptively at three levels: the overall response distribution, the individual meta-requirement level, and the category level. This type of item-level and aggregated analysis has been used in prior practitioner surveys [122]. At the overall level, we calculated the frequencies and percentages of all five response options. At the meta-requirement level, each of the 43 literature-derived meta-requirements was analysed using the 128 corresponding participant ratings. At the category level, ratings were aggregated across the meta-requirements belonging to each of the 16 survey categories; these percentages summarise all ratings within a category and are not respondent-level agreement rates.

For each meta-requirement and category, we reported response frequencies and percentages for agreement, neutral, and disagreement, as well as the median and interquartile range (IQR). Because individual Likert items are ordinal, the median and IQR were used as the primary descriptive summaries [2, 121]. Mean, standard deviation (SD), and mode were reported only as supplementary descriptive measures. Consistent with prior practitioner-survey studies, the quantitative ratings were interpreted alongside open-text feedback to assess support for the proposed requirements and identify refinements or missing concerns [29, 121].

Ratings of 4 (*agree*) and 5 (*strongly agree*) were grouped as agreement-level responses, a rating of 3 as neutral, and ratings of 1 (*strongly disagree*) and 2 (*disagree*) as disagreement-level responses. To provide a consistent descriptive interpretation of practitioner support, we defined study-specific quantitative-status rules for each meta-requirement. Prior consensus studies have used percentage-agreement thresholds to distinguish levels of agreement, including 60% as a minimum consensus threshold and 75% as a commonly used consensus threshold [25, 92]. A meta-requirement was classified as *strongly supported* when its median was at least 4 and at least 75% of ratings were agreement-level responses, and as *supported* when its median was at least 4 and agreement was at least 60% but below 75%. A meta-requirement was classified as *mixed or unclear* when its median was 3 or at least 40% of ratings were neutral, and as *challenged* when its median was at most 3 and at least 30% of ratings were disagreement-level responses. Cases not meeting these criteria were designated as requiring qualitative interpretation. These thresholds were analytic decision rules defined for this study, not inferential significance criteria. The quantitative results were interpreted together with the open-text comments to determine the extent to which the literature-derived meta-requirements were supported, refined, challenged, or extended [29, 121]. No inferential comparisons between demographic subgroups were performed because the survey was designed to evaluate the meta-requirements rather than test differences between respondent groups.

Table 7. Meta-requirements for AI robustness auditing: 43 literature-derived requirements and MR-8.4, added after the practitioner survey.

ID	Sub-Category	Requirement	Source(s)
Human Dimension			
Category 1: Standards and Governance			
MR-1.1	Standardized Metrics	Robustness auditing should use standardised robustness metrics and benchmark definitions where available, so that reported robustness results are comparable across systems, audits, and regulatory contexts.	[1, 9, 14, 19, 22, 30, 48, 51, 55, 69, 72, 73, 85, 89, 90, 99, 116, 117, 127, 130, 131]
MR-1.2	Standardized Lexicon and Terminology	Robustness audits should use a shared vocabulary, taxonomy, and terminology for robustness and auditability concepts.	[19, 22, 30, 36, 48, 55, 69, 72, 80, 85, 89, 99, 116, 117, 120, 127, 130, 131]
MR-1.3	Standardized Tools	Robustness audits should use common or comparable tools, checklists, templates, scorecards, libraries, or toolkits where they support consistent assessment.	[1, 14, 22, 24, 36, 51, 55, 60, 69, 83, 89–91, 99, 116, 117, 130, 131]
MR-1.4	Actionable Guidance	Practitioners should receive practical guidance for selecting tests, thresholds, evidence, mitigations, and audit steps.	[9, 16, 19, 21, 40, 55, 60, 61, 64, 76, 83, 86, 103, 116, 129]
MR-1.5	Robustness-aware Integration Guidance	Robustness guidance should support embedding robustness into development, deployment, monitoring, and assurance workflows.	[14, 19, 22, 30, 48, 51, 55, 60, 69, 72, 73, 89, 90, 99, 116, 117, 127, 131]
MR-1.6	Governance Protocols	Organisational or regulatory protocols should define responsibility, escalation, remediation, accountability, and risk decisions for robustness assurance.	[15, 21, 22, 40, 64, 70, 72, 76, 82, 89, 99, 116]
MR-1.7	Audit Access Policy	Audit access policy should specify the artefacts and access level required for robustness assessment, including relevant datasets, documentation, source code, model or system information, deployment details, logs, internal evaluation results, APIs, and conditions for black-box, white-box, or outside-the-box access.	[14, 16, 22, 39, 61, 70, 84, 91, 99, 110, 116]
MR-1.8	Auditor Protocols	Auditor protocols should specify how the audit is conducted, including auditor conduct, independence, method transparency, evidence handling, reporting format, and conditions under which audit results may be interpreted.	[14, 19, 22, 30, 36, 55, 60, 69, 72, 73, 82, 85, 89–91, 99, 116, 117]
Category 2: Engagement			
MR-2.1	Diverse Stakeholders	Robustness audits should identify and involve relevant technical, domain, governance, user, and affected-party perspectives, including users or groups exposed to robustness failures, according to the robustness risks and impacts covered by the audit scope.	[7, 16, 22, 24, 39, 40, 42, 48, 52, 64, 65, 70, 72, 73, 76, 82–85, 89, 91, 99, 101, 108, 117, 133]
MR-2.2	Stakeholder Involvement Prioritization	Stakeholder involvement should be prioritised according to each group’s exposure to robustness failures, domain knowledge, decision authority, and relevance to deployment risks.	[1, 22, 60, 69, 73, 89, 90, 99, 116]
MR-2.3	Independent Auditor	Robustness audits should include independent review with conflict control, methodological transparency, and sufficient access for the intended assurance claim.	[14, 22, 40, 60, 64, 70, 72, 76, 82, 91, 99, 116]
MR-2.4	Early-stage Involvement	Domain and technical experts should be involved during problem framing, design, and assessment planning when early expert judgement is needed to define robustness-relevant assumptions, risks, or evidence.	[21, 48, 52, 70, 73, 76, 99]
Category 3: Robustness Definition			
MR-3.1	Scoping Robustness Concerns	The robustness scope should define the audit claim, system boundary, disturbance types, failure modes, deployment contexts, assumptions, and justified exclusions.	[1, 7–9, 14–16, 19, 20, 22, 24, 30, 36, 39–41, 48, 51, 53–55, 61, 65, 69, 72, 76, 80, 82, 86, 89, 90, 93–96, 99, 101–103, 105, 108, 110, 113, 114, 117, 118, 120, 125, 127, 129, 130, 133]
MR-3.2	Robustness Trade-off Assumptions	The audit should identify assumptions about how the scoped robustness claim may affect, depend on, or conflict with other system properties, including accuracy, fairness, privacy, explainability, safety, cost, and affected-user outcomes, and should document how these assumptions shape measurement, interpretation, and decision-making.	[7, 8, 16, 19, 20, 22, 24, 30, 36, 40, 41, 43, 51–54, 61, 65, 67, 69, 70, 72, 73, 76, 80, 82, 85, 89, 90, 93–95, 103, 110, 114, 116–118, 120, 130]
Category 4: Knowledge and Skills			
MR-4.1	Multidisciplinary Skills	Robustness audits should involve combined AI, machine-learning, software, security, data, domain, social, legal, risk, and audit expertise where required by the assessment.	[4, 22, 52, 55, 60, 64, 69, 70, 73, 76, 82, 84, 89, 101, 108, 117, 120, 129]
MR-4.2	Robustness-awareness Educational Programmes and Training	Personnel should receive education or training on robustness risks, methods, monitoring, mitigation, communication, and tool use relevant to their roles.	[4, 16, 37, 40, 60, 61, 84, 133]
Functional Dimension			
Category 5: Data Auditability			
MR-5.1	Data Coverage and Sufficiency	Robustness assessment data should sufficiently cover relevant populations, data types, modalities, subgroups, scenarios, domains, edge cases, and operating conditions.	[1, 19, 22, 30, 48, 51, 60, 69, 73, 85, 89, 90, 99, 117, 127, 130, 131]

(continued on next page)

(continued from previous page)

ID	Sub-Category	Requirement	Source(s)
MR-5.2	Data Description, Provenance, and Lineage	Robustness audits should require dataset-level descriptions and provenance records that identify data sources, acquisition conditions, processing steps, annotation basis, known limitations, and links between the data and the robustness claim.	[8, 14, 22, 36, 51, 55, 60, 67, 69, 89, 90, 99, 116, 127]
MR-5.3	Data Quality	Data used for robustness assessment should be checked for noise, missingness, errors, corruption, leakage, imbalance, representativeness, and other data defects.	[8, 9, 43, 67, 75, 76, 80, 86, 90, 93, 108, 130]
MR-5.4	Data Subgroup and Bias Source Identification	Data subgroups, target attributes, representation gaps, and data-level bias sources should be identified where they may affect the credibility of robustness findings across relevant populations, contexts, or operating conditions.	[5, 52, 54, 55, 60, 69, 76, 82, 90, 102]
MR-5.5	Robustness Pre-processing Mitigation	The audit should assess whether data-level mitigation methods, such as cleaning, filtering, transformation, augmentation, or recovery of contaminated training data, address identified robustness risks and improve robustness under the assessed conditions.	[9, 19, 24, 30, 51, 69, 94, 117, 127]
Category 6: Model Auditability			
MR-6.1	Model Robustness Probing	Auditors should probe model behaviour with defined robustness stressors, including perturbations, adversarial examples, stress cases, distribution shifts, benchmark scenarios, and complementary test methods where one probing method is insufficient.	[1, 22, 24, 36, 42, 43, 53, 62, 65, 75, 82, 85, 89, 90, 93, 96, 101, 105, 108, 117, 118, 120, 125, 130]
MR-6.2	Model Intended Use	The audit's intended application, users, operating conditions, automation level, limitations, and unsuitable uses should be specified for the audited model.	[73, 89, 90, 99, 116]
MR-6.3	Model Explainability	Model behaviour, decision logic, and robustness failures should be interpretable or otherwise explainable enough to support the audit.	[4, 14–16, 20, 21, 36, 40, 42, 51–55, 60, 61, 69, 70, 72, 76, 78, 80, 90, 93, 95, 96, 102, 103, 105, 113, 114, 117, 133]
MR-6.4	Model Bias and Failure Source Identification	Model-level causes of robustness failure, including shortcut learning, overfitting, spurious correlations, and unstable features, should be identified where relevant.	[14, 30, 42, 54, 65, 96, 101]
MR-6.5	Robustness In-processing Mitigation	The audit should assess whether training-time mitigation methods, such as adversarial training, robust optimisation, regularisation, or adapted loss functions, address identified robustness risks and improve robustness under the assessed conditions.	[1, 51, 117, 127, 130]
Category 7: Output Auditability			
MR-7.1	Robustness Post-processing Mitigation	Robustness audits should identify output-level unstable, unsafe, biased, harmful, or inconsistent behaviour under robustness stress, trace the likely source of the failure, and assess whether post-processing mitigations reduce the identified robustness risk before or during deployment.	[19, 51, 72, 117, 131]
Category 8: Workflow Integration			
MR-8.1	Software Development Process Integration	Robustness checks should be integrated into requirements, design, implementation, testing, deployment, maintenance, and review activities.	[5, 9, 19, 52, 64, 69, 75, 80, 108, 114]
MR-8.2	MLOps Frameworks Integration	Robustness tests, alerts, retraining triggers, monitoring, and reassessment steps should be embedded into ML or MLOps pipelines where applicable.	[8, 24, 36, 61, 86, 127]
MR-8.3	Robustness-aware Integration Compatibility	Robustness tools and procedures should be compatible with relevant software, data, model, deployment, monitoring, and incident-management infrastructure.	[24, 55, 60, 61, 75, 116, 117]
MR-8.4*	External Component and Supply Chain Assurance	Robustness audits should identify external models, data, software, services, and infrastructure on which the audited AI-enabled system depends; assess how their failure or change could affect system behaviour; and state how limited access to those dependencies restricts the robustness conclusion.	Practitioner feedback (P079, P102); supported by prior studies [9, 14, 24, 75, 82, 131]
Category 9: Human-in-the-Loop			
MR-9.1	Robustness-aware Human Participation Levels	Robustness audits should define human participation levels for robustness-relevant decisions, including review, validation, intervention, override, or approval, and should state how available access to inputs, outputs, procedures, transparency information, model internals, or other evidence limits the resulting robustness judgement.	[14, 15, 19, 40, 52, 60, 62, 76, 90, 116, 117]
Category 10: Evaluation			
MR-10.1	Continuous Robustness Assessment	Robustness should be formally reassessed repeatedly or when triggered by changes in data, users, attacks, models, or deployment conditions.	[9, 19, 22, 55, 61, 64, 70, 73, 76, 82, 83, 99, 110, 116, 125]
MR-10.2	Robustness-aware Progress Tracking and Prioritisation	Robustness evaluations should maintain a prioritised record of completed checks, unresolved robustness concerns, follow-up tests, mitigation needs, and reassessment actions, so that remaining audit work is visible and sequenced.	[19, 22, 24, 82, 117]
MR-10.3	Cross-Layer Failure Validation	Selected failures should be validated across relevant data, model, software, service, infrastructure, and human-interaction layers.	[9, 24, 80, 99, 131]
Category 11: Monitoring			

(continued on next page)

(continued from previous page)

ID	Sub-Category	Requirement	Source(s)
MR-11.1	Continuous, Automated, and Real-time Robustness Monitoring	Deployed AI-enabled systems should maintain a defined post-deployment robustness monitoring process that tracks robustness indicators, applies predefined thresholds, issues automated or real-time alerts for specified robustness violations, and triggers reassessment when system behaviour, data, deployment conditions, or threat conditions change.	[4, 8, 9, 15, 19, 20, 39–41, 53, 55, 61, 70, 75, 76, 78, 80, 83, 86, 89, 91, 99, 101, 102, 106, 116, 120]
MR-11.2	Robustness-aware Logging	Inputs, outputs, model behaviour, errors, alerts, updates, and context should be logged where needed to diagnose robustness failures.	[70, 78, 99, 106, 116]
Category 12: Communication			
MR-12.1	Robustness Dashboards and Visualisation	Robustness dashboards and visualisations should present audit-relevant metrics, model behaviour, monitoring trends, subgroup outcomes, failures, alerts, and supporting evidence in a form that helps intended users inspect and interpret robustness evidence.	[1, 4, 60, 70, 78, 84, 89, 90, 99, 101, 117, 129]
MR-12.2	Robustness Documentation and Risk Report	Robustness audits should document and report the audit objective, evidence base, evaluated system and datasets, robustness findings, limitations, mitigations, residual risks, reproducibility conditions, traceability information, and audience-appropriate explanations needed to interpret the audit conclusion.	[4, 5, 8, 9, 14, 16, 19, 21, 22, 24, 30, 39, 42, 53, 54, 61, 64, 69, 70, 76, 78, 82, 83, 90, 91, 93, 94, 96, 99, 110, 113, 118, 120, 131, 133]
Quality Dimension			
Category 13: Efficiency			
MR-13.1	Resource Efficiency and Scalability	A robustness auditing process should be cost-effective, time-efficient, and scalable, so that organisations with different resource capacities can assess, monitor, mitigate, and report robustness risks within practical audit-cycle constraints and across the models, datasets, perturbations, users, and deployment settings covered by the audit scope.	[20, 24, 39, 48, 51, 53, 55, 60, 62, 73, 75, 76, 80, 82, 85, 105, 116, 117, 125, 127]
Category 14: Adaptability			
MR-14.1	Domain-specific Adaptability	Robustness definitions, perturbations, thresholds, tests, and interpretations should be tailored to risks and evidence standards specific to the deployment domain.	[7, 14, 20, 30, 39, 43, 51, 52, 55, 67, 80, 82, 84, 90, 93, 96, 102, 117, 125]
MR-14.2	Task and Model Versatility	Robustness assessment should support the task types, model classes, modalities, datasets, and deployment settings covered by the audit scope.	[1, 7, 51, 54, 69, 116]
Category 15: Accessibility			
MR-15.1	Accessible and User-Friendly Robustness Tools, Visualisations, and Documents	Robustness tools, visualisations, documents, and outputs should be understandable, usable, and available at an appropriate level of technical detail for their intended technical and non-technical audiences.	[1, 16, 19, 22, 30, 42, 51, 54, 60, 69, 70, 73, 76, 78, 89, 91, 131]
Category 16: Configurability			
MR-16.1	Configurable Robustness Assessment Tools	Robustness assessment tools should allow configuration of perturbations, thresholds, metrics, dashboards, reports, tools, and deployment-specific scenarios.	[1, 30, 51, 60, 72, 94, 120, 130]

Note. *MR-8.4 was not identified during the initial synthesis of the 74 retained studies; it emerged from practitioner feedback and was subsequently supported by evidence found on re-examining those studies.

4.4.6 Qualitative Coding of Open-Text Comments. The 586 substantive open-text comments were organised as separate records and linked to an anonymous respondent identifier (P001-P128), source category, and relevant meta-requirement. Each comment was manually coded for its theme, function, and related meta-requirement, with supporting quotations and coder notes retained for traceability. Comment functions captured confirmations, clarifications, implementation concerns, challenges, and proposed additions, while coding decisions distinguished retention, refinement, addition, merging or splitting, and no change. Examples of anonymised comments and their Retain, Refine, Add, and Not usable coding decisions are provided in the replication package (see the Data Availability statement). To reduce dependence on a single interpretation, one researcher coded the comments, while a second researcher independently reviewed the coding and proposed synthesis decisions, following multi-researcher coding practices used in related practitioner studies [121, 123]. We then reconciled the qualitative feedback with the quantitative results in Phase 4. For each meta-requirement, the quantitative support status was considered together with the related practitioner comments. Proposed changes were compared against the complete set of meta-requirements to determine whether they supported

retention, required refinement, or identified a concern not already represented in the existing set. This integration informed the final practitioner-informed set of meta-requirements.

4.5 Phase 4: Meta-Requirement Synthesis

We integrated the quantitative ratings and qualitative comments to produce the final practitioner-informed set of meta-requirements. For each literature-derived meta-requirement, we considered the quantitative support status alongside the associated qualitative feedback. Practitioner suggestions were then compared with the existing meta-requirements and relevant literature to determine whether they reinforced, refined, challenged, or extended the literature-derived set of meta-requirements. Qualitative synthesis decisions were reconciled as *Retain*, *Refine*, *Add*, *No change*, or *Not usable*. We retained suggested additions only when the underlying concern was not sufficiently represented by an existing meta-requirement and was supported by relevant evidence; otherwise, we incorporated them as refinements or did not adopt them. Merge and split suggestions were similarly reviewed against neighbouring meta-requirements and supporting evidence to clarify requirement boundaries where appropriate. The synthesis retained the 43 literature-derived meta-requirements, refined their wording and scope where supported by practitioner feedback and the evidence base, and added one practitioner-informed requirement. The final set therefore comprises 44 meta-requirements across the original 16 categories.

MR-8.4, *External Component and Supply Chain Assurance*, was added to Category 8, Workflow Integration, because two overall survey comments (P079, P102) identified a concern not covered by the existing meta-requirements. Comparison against the complete set showed that MR-8.1-MR-8.3 were the closest matches, but they did not address limited auditor access to external dependencies, the use of third-party validation evidence, or how these limitations affect the conclusions that can be drawn. MR-8.4 was therefore retained as a separate meta-requirement and placed in Workflow Integration because these external dependencies form part of the AI-enabled system’s operational workflow. As MR-8.4 was introduced after the survey, it was not rated in the original Likert-scale validation and therefore does not share the quantitative validation status of the other 43 meta-requirements.

5 Meta-Requirements for AI Robustness Auditing

This section presents the final practitioner-informed meta-requirements for AI robustness auditing (RQ1). As shown in Figure 5, the final set consists of 44 meta-requirements across 16 categories and three higher-level dimensions: Human (16 meta-requirements), Functional (23), and Quality (5). We derived the initial set of 43 meta-requirements from the 74 retained studies and refined it with practitioner feedback, including the addition of MR-8.4, *External Component and Supply-Chain Assurance*. The replication package (see the Data Availability statement) reports the original 43 literature-derived meta-requirements used in the survey. The human dimension covers the organisational, stakeholder, conceptual, and competence conditions needed to make robustness auditing governable and interpretable.

The quality dimension covers the qualities that make robustness auditing feasible, context-sensitive, understandable, usable, and configurable across different systems and deployment settings. Finally, the functional dimension covers the audit activities and evidence needed across the AI-enabled system lifecycle, including data, model, output, workflow, human review, evaluation, monitoring, and communication. The rest of this section presents each meta-requirement and category.

Human Dimension						
① Standards & Governance			② Engagement			
Standards	Guidance	Policies	Stakeholder	User	Auditor	Expert
- Standardized Metrics - Standardized Lexicon & Terminology - Standardized Tools	- Actionable Guidance - Robustness-aware Integration Guidance	- Governance Protocols - Audit Access Policy - Auditor Protocols	- Diverse Stakeholders - Stakeholder Involvement Prioritization	Identify Affected Users	Independent Auditor	Early-stage Involvement
③ Robustness Definition			Knowledge & Skills			
Scoping Robustness Concerns		Robustness Trade-off Assumptions	- Multi-disciplinary Skills - Robustness-awareness Educational programs/Trainings			
Functional Dimension						
⑤ Data Auditability		⑥ Model Auditability		⑦ Output Auditability		
- Data Coverage and Sufficiency - Data Description, Provenance, and Lineage - Data Quality - Data Subgroup and Bias Source Identification - Robustness Pre-processing Mitigation		- Model Robustness Probing - Model Intended Use - Model Explainability - Model Bias and Failure Source Identification - Robustness In-processing Mitigation		Robustness Post-processing Mitigation		
⑧ Workflow Integration		⑨ Human-in-the-Loop		⑩ Evaluation		
- Software Development Process Integration - MLOps Frameworks Integration - Robustness-aware Integration Compatibility - External-Component and Supply-Chain Assurance		Robustness-aware Human Participation Levels		- Continuous Robustness Assessment - Robustness-aware Progress Tracking and Prioritisation - Cross-Layer Failure Validation		
⑪ Monitoring		⑫ Communication				
- Continuous, Automated, and Real-time Robustness Monitoring - Robustness-aware Logging		Visualization		Documentation & Report		
		Robustness Dashboards and Visualisation		Robustness Documentation and Risk Report		
Quality Dimension						
⑬ Efficiency		⑭ Adaptability	⑮ Accessibility		⑯ Configurability	
Resource Efficiency and Scalability		- Domain-Specific Adaptability - Task and Model Versatility	Accessible and User-Friendly Robustness Tools, Visualisations, and Documents		Configurable Robustness Assessment Tools	

Fig. 5. Final practitioner-informed meta-requirements for AI robustness auditing, comprising 44 meta-requirements across 16 categories and three dimensions.

5.1 Category 1: Standards and Governance

This category defines the measurement conventions, terminology, tools, guidance, governance responsibilities, access conditions, and audit procedures needed to make AI robustness audit evidence comparable, interpretable, and governable.

5.1.1 MR-1.1: Standardised Metrics. Standardisation means using the same measures and the same rules for deciding whether a result is acceptable when similar robustness claims are assessed. Audit results are difficult to compare when similar claims are tested using different measures, test conditions, or acceptance rules [9, 89, 117]. Prior studies measure robustness through performance under corrupted data, adversarial attacks, and changes in the data distribution [14, 19, 22, 55, 85, 90, 99, 116, 130]. Each audit should therefore state what is measured, how it is measured, the conditions under which it is tested, and what result is considered acceptable [1, 9, 73, 89, 131].

5.1.2 MR-1.2: Standardised Lexicon and Terminology. A common set of definitions helps everyone involved in the audit use key robustness terms consistently. This is necessary because terms such as robustness, reliability, failure, and auditability can have different meanings in engineering, legal, and organisational work [30, 48, 80, 117, 120, 127, 131]. If these terms are not defined clearly, auditors, developers, regulators, and system users may interpret the same robustness claim differently [22, 99, 116]. The audit should therefore define the terms needed to understand the claim, the assessment, and the conclusion, and should distinguish robustness from related properties such as safety, fairness, privacy, explainability, and accountability [19, 36, 55, 69, 72, 80, 85, 89, 120, 130].

5.1.3 MR-1.3: Standardised Tools. This MR emphasizes using software tools or test instruments in a consistent way to assess the same type of robustness problem. Different tools may be used for the same type of assessment, provided that they conform to a common standard that supports consistent and comparable results [1, 24, 83, 90, 91, 99, 131]. Tools differ in what they test, the conditions they can reproduce, and how they are configured, so a tool should be selected to match the robustness claim, the system or component being assessed, and the type of input data [14, 36, 51, 55, 117, 130]. The audit should state which tool and version were used, the main settings applied, and any known limitations that affect the result [14, 60, 116].

5.1.4 MR-1.4: Actionable Guidance. This MR explains how a robustness requirement is turned into specific tasks that an auditor can carry out for the system being assessed [55, 60]. The guidance should translate applicable technical, legal, and organisational requirements, including requirements related to properties such as safety, security, or reliability where they are relevant, together with conditions that could cause the system to fail or be deliberately manipulated, into concrete audit actions [21, 40, 86, 116, 129]. It should state which checks or tests to perform, what evidence to examine, and when further review is required [9, 76].

5.1.5 MR-1.5: Robustness-aware Integration Guidance. The guidance on when and how robustness checks should be included as an AI-enabled system is developed, deployed, used, changed, and reviewed, rather than being performed only as a separate test [19, 69, 73, 99, 116, 117, 131]. The guidance should state at which stages the checks are required and when they must be repeated after changes to the model, data, operating conditions, or intended use of the AI-enabled system [48, 51, 55, 60, 72, 89, 90, 127]. MR-1.5 concerns when and where robustness checks should be required across the AI lifecycle, whereas MR-8.1 concerns whether those checks are integrated into development workflows.

5.1.6 MR-1.6: Governance Protocols. These protocols define who is responsible for investigating robustness problems, deciding what action is needed, and deciding whether the AI-enabled system may continue to be used when a problem cannot be fully resolved [21, 70, 76]. Prior work assigns these responsibilities to organisations that develop, operate, or oversee AI-enabled systems [40, 72, 82, 99]. The protocols should state who investigates the problem, who decides on corrective action, who approves continued use of the system, and who must be informed when the problem could affect important properties of the AI-enabled system or its outcomes, such as safety, security, fairness, reliability, or privacy, or could result in non-compliance with applicable law [15, 22, 64, 72, 82, 89, 116].

5.1.7 MR-1.7: Audit Access Policy. This policy defines what information and system components the auditor is allowed to inspect and what remains unavailable. Audit conclusions may depend on access to data, source code, model information, system documentation, logs, test results, and deployment information [14, 16, 61, 70, 84, 91, 99]. The policy should state what can and cannot be inspected, how long that access is available, and how any restrictions limit the conclusions that can be drawn [22, 61, 84, 91, 110, 116]. When privacy, security, legal, or confidentiality requirements prevent access to some information, the policy should identify the restriction and any permitted alternative source of information [16, 110, 116].

5.1.8 MR-1.8: Auditor Protocols. These protocols define how a robustness audit is carried out, including who performs it, the steps followed, the methods used, and how the results from those methods are used to reach the final conclusion. Audits may be conducted by internal teams or independent external assessors and may include technical testing, document review, or examination of how the organisation develops and operates the system [14, 19, 36, 60, 116]. The protocols should state the audit objective, the order of the audit activities, the methods and criteria used, and what

each method contributes to the final conclusion [14, 22, 90, 91]. When more than one method is used, the protocol should explain how their results are combined to support that conclusion [72, 73, 85]. MR-1.8 concerns how the audit is conducted, whereas MR-2.3 concerns the role, independence, and access of an independent auditor.

5.2 Category 2: Engagement

This category establishes which stakeholder, user, expert, and independent-auditor perspectives must inform the framing, assessment, and use of AI robustness audit evidence.

5.2.1 MR-2.1: Diverse Stakeholders. The stakeholders include individuals and organisations that develop, operate, use, oversee, or experience consequences from decisions made with an AI-enabled system or from its failures. An audit that relies only on technical developers may miss robustness problems, harms, or practical constraints recognised by other stakeholders [7, 16, 22, 70, 76, 99, 101]. The audit should therefore include perspectives from the stakeholder types connected to the robustness claim, especially where the same failure may have different consequences for different people [42, 72, 82, 85, 89, 117, 133].

5.2.2 MR-2.2: Stakeholder Involvement Prioritisation. This MR determines which stakeholders need more involvement in the audit, when their input is needed, and how much participation is required. Priority should depend on how directly they may experience the consequences of a system failure, the knowledge they can contribute, and the likelihood and seriousness of the failure being assessed [22, 69, 73, 89, 90, 99, 116]. The audit should therefore state which stakeholders need one-time, repeated, or focused involvement and why, while considering available resources and any restrictions on access to the system or information [69, 90, 99, 116].

5.2.3 MR-2.3: Independent Auditor. Auditor independence means that the person or organisation conducting the audit can reach a conclusion without pressure from parties that develop, operate, fund, or manage the system being assessed [76, 91]. External auditors may be more separate from the organisation, while internal auditors may have more detailed knowledge of the system and its development [14, 72, 99]. The audit should therefore state who selected and funds the auditor, whether the auditor previously helped develop or operate the system, and whether any organisational relationship could influence the audit conclusion [22, 64, 70, 76, 82, 116]. Any relationship that could influence the auditor’s judgement should be disclosed and managed [70, 76].

5.2.4 MR-2.4: Early-stage Involvement. This MR implies bringing the needed expertise early in planning the robustness audit, such as technical, domain, legal or compliance, and organisational expertise, before key audit decisions are made. These decisions may include which robustness problems will be examined, which parts of the AI-enabled system will be included, which tests will be performed, what evidence will be collected, and what results will be considered acceptable. This is important because robustness problems can be harder to address when the needed expertise is introduced only after the audit has already been planned or testing has begun [99]. Experts should therefore contribute while the audit scope, test plan, evidence needs, and acceptance criteria are being decided [21, 48, 70, 73, 76]. Their early input should help ensure that the audit examines the robustness problems and system conditions that matter for the case being audited [21, 70, 73, 76].

5.3 Category 3: Robustness Definition

This category defines the robustness claim an audit will assess, the assumptions that bound that claim, and the trade-offs that affect how robustness evidence is interpreted.

5.3.1 MR-3.1: Scoping Robustness Concerns. This MR considers which robustness problems the audit will assess, the conditions under which they will be tested, and which conditions are excluded. The audit should state whether it covers adversarial robustness, non-adversarial robustness, or both (see Table 1 for definitions) [8, 19, 30, 48, 69, 72, 117, 125, 133]. It should then identify the specific conditions to be tested, such as corrupted or missing data, distribution changes, poisoning, or evasion attacks [7, 9, 16, 22, 51, 61, 82, 86, 95, 103, 105, 114, 120, 127].

5.3.2 MR-3.2: Robustness Trade-off Assumptions. A robustness trade-off occurs when improving robustness changes another important property of the system, such as accuracy, computational cost, explainability, fairness, safety, and privacy. Ignoring such changes can make a robustness improvement appear beneficial even when another property becomes worse [36, 61, 72, 85]. The audit should therefore identify which properties may change, compare them before and after the method used to improve robustness, and state how any observed gain or loss affects the robustness conclusion [8, 20, 51, 69, 94, 110, 117, 120, 130].

5.4 Category 4: Knowledge and Skills

This category defines the competence and role-specific training needed to conduct, interpret, and maintain AI robustness audits.

5.4.1 MR-4.1: Multidisciplinary Skills. These mean having expertise from more than one field when a robustness audit requires different kinds of knowledge. This is important because assessing robustness may require knowledge of AI models, the software system in which they operate, the domain in which the system is used, and relevant security, legal, or organisational requirements [70, 82, 129]. The audit should therefore involve people with the skills needed for the system and robustness problems being assessed, which may include machine learning, software engineering, cybersecurity, statistics, law or policy, auditing, and domain expertise [22, 55, 64, 89]. If required expertise is missing, the audit should state how that limitation affects the conclusions that can be drawn [4, 60, 76, 84, 108, 120].

5.4.2 MR-4.2: Robustness-awareness Educational Programmes and Training. Training should give each person the knowledge needed for their role in robustness auditing. Auditors, developers, system administrators, users, and domain experts may need different training because they perform different tasks and make different decisions [16, 133]. The training should cover the robustness problems, methods, tools, and responsibilities relevant to each role, including adversarial testing or secure system operation where needed [4, 37, 40, 60, 61, 84]. Because methods, tools, attacks, and legal requirements change over time, the training should be updated periodically and may include practical exercises, simulated failures, or audit examples [84, 133].

5.5 Category 5: Data Auditability

This category establishes whether data used in robustness audits are sufficiently covered, documented, reliable, bias-aware, and prepared for data-level mitigation before interpreting robustness claims.

5.5.1 MR-5.1: Data Coverage and Sufficiency. Data coverage means that the data used in the audit include the people, cases, and operating conditions covered by the robustness claim. Beyond including these cases and conditions, the audit also needs enough data to assess them reliably; this is what we mean by data sufficiency. Robustness evidence can be misleading when important groups, rare cases, distribution changes, or failure conditions are missing, or when too little data are available to assess them reliably [51, 73, 85]. The audit should therefore check whether the available data represent the intended users, important variations in the data, rare but important cases, and the conditions under

which the AI-enabled system is expected to operate, and whether enough data are available to evaluate these reliably [1, 19, 22, 30, 48, 69, 89, 90, 117, 127, 130, 131]. If important cases or conditions are missing, or are represented by too little data, the audit should state how these limitations affect the robustness conclusion [60, 99].

5.5.2 MR-5.2: Data Description, Provenance, and Lineage. This MR describes what the data contain, where they came from, and how they were collected or changed before being used in the audit. Without this information, auditors may not know whether the history of the data influenced the robustness result [99, 116]. The audit should identify the data source, collection conditions, labelling process, preprocessing steps, dataset versions, and any synthetic data used [8, 67, 90]. It should also trace major changes made to the data used in the audit, such as filtering, merging, relabelling, or transformation, so that their origin and processing history can be checked [22, 36, 60, 90, 99, 116].

5.5.3 MR-5.3: Data Quality. This concerns whether the data used in the audit are accurate, consistent, and suitable for producing reliable robustness results. Errors such as incorrect labels, duplicated records, corrupted inputs, missing values, data leakage, or contamination can distort the assessment [8, 43, 108]. The audit should therefore check for these problems and explain how any detected defects affect the robustness conclusion [9, 43, 75, 76, 80, 86, 93, 130]. Where reference labels are used, their accuracy and completeness should also be checked.

5.5.4 MR-5.4: Data Subgroup and Bias Source Identification. This MR examines whether robustness differs across defined groups in the data and whether those differences can be linked to how the data were collected, labelled, or processed. Overall results can hide weaker performance for particular demographic, geographic, device, usage, or domain groups [5, 54, 69, 76, 82, 90, 102]. The audit should therefore compare robustness across the groups relevant to the claim and investigate whether observed differences are associated with sampling imbalance, annotation errors, preprocessing choices, missing group information, or changes in the data distribution [54, 55, 60]. These data-related causes should be considered when interpreting differences in robustness across groups [5, 102].

5.5.5 MR-5.5: Robustness Pre-processing Mitigation. Pre-processing robustness mitigation refers to data-level changes made before model training or evaluation to address an identified robustness problem. Examples include cleaning corrupted data, filtering harmful inputs, transforming features, balancing data, or generating additional test or training examples [19, 24, 30, 94, 117, 127]. The audit should assess whether the chosen data changes resolve the identified problem and whether they actually improve robustness under the conditions being tested [24, 51, 117]. It should also check whether those changes hide existing weaknesses or introduce new ones [127].

5.6 Category 6: Model Auditability

This category establishes how model behaviour, intended use, explanations, failure sources, and training-time mitigation should be made auditable for robustness claims.

5.6.1 MR-6.1: Model Robustness Probing. This MR means deliberately testing the model under the robustness conditions selected in MR-3.1 to see how its behaviour changes. This is needed because a model may appear to perform well overall while still failing under specific conditions [43, 53, 75, 96, 101, 120]. The audit should use tests that match the robustness problem being assessed, such as corrupted inputs, distribution changes, specially constructed test cases, or adversarial examples [1, 24, 36, 65, 89, 90, 117, 118, 125, 130]. For each test, the audit should state the test conditions, the measure used, and how the result is interpreted [82, 93].

5.6.2 MR-6.2: Model Intended Use. This defines what the model is designed to do, who is expected to use it, and the operating conditions under which its robustness is assessed. Robustness results can be misleading if they are applied to uses or conditions that were not examined [73, 90, 99, 116]. The audit should therefore state the model’s intended task, expected users, role in decision-making, operating conditions, known limitations, and uses that fall outside the assessment [73, 89, 99]. This keeps the robustness conclusion limited to the uses and conditions that were actually evaluated [90, 116].

5.6.3 MR-6.3: Model Explainability. This MR encompasses using methods that help auditors understand why a model produced a particular result or behaved differently under a robustness test. This is useful when the model’s behaviour cannot be understood from performance results alone [55, 76, 80]. Such methods may show which inputs or model features influenced the result and may help reveal errors, bias, or unusual behaviour [16, 40, 51, 72, 78, 93, 114, 117, 133]. The audit should use methods that show why the model produced a result only when they help answer the robustness question being assessed, and should state what those methods can and cannot show [4, 14, 15, 21, 53, 54, 60, 61, 69, 70, 90, 95]. Because such explanations can be unstable or misleading, they should not be treated as proof of the true cause of a model failure [14, 36, 42, 53, 105, 113].

5.6.4 MR-6.4: Model Bias and Failure Source Identification. This examines which parts of the model’s learned behaviour may be responsible for an observed robustness weakness. A model may perform poorly because it relies on misleading patterns, unstable features, memorised examples, or rules that do not hold outside the data used during development [14, 30, 42, 96, 101]. The audit should therefore investigate whether the weakness can be linked to the features the model relies on, the patterns it has learned, or differences in behaviour across tested cases, rather than reporting only the performance loss [42, 54, 65, 101]. Any claimed cause should be supported by evidence from the tests performed [54].

5.6.5 MR-6.5: Robustness In-processing Mitigation. This mitigation method changes how the model is trained so that it performs more reliably under the failure condition being assessed. Such changes may include training with adversarial examples, adjusting the training objective, or using methods that reduce reliance on patterns that fail when the input or operating conditions change [1, 51, 117, 127]. The audit should assess whether the chosen training change addresses the robustness problem being tested and whether the model performs more reliably after training [1, 51]. Any improvement should be verified through testing rather than assumed from the use of the training method alone [130].

5.7 Category 7: Output Auditability

This category examines how robustness audits connect observed outputs and inference behaviour to the failure sources that produced them.

5.7.1 MR-7.1: Robustness Post-processing Mitigation. Post-processing robustness mitigation refers to measures applied after model inference to reduce the effect of an identified robustness problem. Examples include rejecting uncertain outputs, checking outputs against additional rules, routing selected cases for human review, or using a safer alternative when an output cannot be trusted [19, 131]. The audit should assess whether the selected mitigation reduces the identified robustness problem under the conditions being tested and should compare system behaviour before and after the mitigation is applied [19, 72, 117, 131]. If the same failures remain, the post-processing mitigation should not be treated as sufficient evidence that the robustness problem has been resolved [51, 72, 117].

5.8 Category 8: Workflow Integration

This category establishes how robustness auditing requirements are embedded into development workflows, machine-learning operations, and existing organisational systems rather than treated as isolated assessment tasks [8, 108, 117].

5.8.1 MR-8.1: Software Development Process Integration. This MR means building robustness checks into the activities used to develop and change the AI-enabled system, rather than performing them only at the end of development. Robustness problems can be introduced when requirements, data, models, code, or system components are changed [5, 64, 75, 80, 108]. The audit should therefore check whether robustness testing is included at the development stages where such changes occur and whether testing is repeated after important changes to the model, data, code, or supporting infrastructure [9, 19, 69, 75, 108, 114].

5.8.2 MR-8.2: MLOps Frameworks Integration. This MR emphasises including robustness checks in the processes used to release, deploy, update, and operate ML models. Robustness problems can appear when models or data change after development, so these processes should not rely only on tests performed before release [8]. The audit should check whether the operational workflow includes robustness testing before release, checks on production data, verification that protective measures still work, and clear actions when required conditions are not met [24, 86, 127]. Those actions may include stopping a release, pausing operation, or returning to an earlier model version [36].

5.8.3 MR-8.3: Robustness-aware Integration Compatibility. Integration compatibility means that the tools used for robustness auditing can work with the system being assessed and its existing technical environment. Audit tools are less useful when they cannot connect to the system, read the required data, or run under its operating constraints [60, 75, 116, 117]. The audit should therefore check whether the selected tools work with the system's interfaces, data formats, software environment, and available computing resources [24, 61, 75, 116]. Using audit tools should not require converting data into unsupported formats, disrupt communication with the system, cause unacceptable delays, or prevent required robustness tests from being completed [55].

5.8.4 MR-8.4: External Component and Supply Chain Assurance. This MR concerns models, data, software, services, or infrastructure supplied by another organisation on which the audited AI-enabled system depends. Practitioner feedback identified this as a missing MR in the initial literature-based MRs: P079 highlighted "third-party and supply chain risks" and restricted auditor access to external components, while P102 called for third-party validation and guidance for auditing foundation models and third-party APIs. Prior studies likewise show that third-party components, APIs, software dependencies, data pipelines, and infrastructure can introduce failures or changes that affect the behaviour of the wider AI-enabled system [9, 24, 75, 131]. They also show that the strength of an audit conclusion depends on the information and system access available to the auditor [14, 82]. The audit should therefore identify external dependencies that can affect the robustness claim, assess how their failure or change could alter system behaviour, and state how any inability to examine them limits the conclusion that can be drawn. To support this assessment, the audit may use Software Bills of Materials (SBOMs), which document software components and dependencies, and Artificial Intelligence Bills of Materials (AIBOMs), which extend this documentation to AI-specific assets such as models, datasets, software dependencies, configurations, and provenance information [88, 98].

5.9 Category 9: Human-in-the-Loop

This category defines how human judgement, review, and oversight are assigned when robustness auditing decisions cannot be supported by automated procedures alone.

5.9.1 MR-9.1: Robustness-aware Human Participation Levels. Human participation defines when a person must review a robustness-related result, what action that person may take, and whether the system can continue operating without human approval. Human review may be needed when a robustness test fails, the system repeatedly produces unreliable results, or continued operation could create serious consequences [14, 15, 19, 60, 62, 76, 117]. The audit should state when human review is required and whether the reviewer may correct a result, reject a system output, approve continued operation, or stop the system from acting [14, 15, 19, 90, 116]. Different forms of human involvement should be distinguished by whether the person only reviews the system, can intervene in its operation, or has final authority over its actions [40].

5.10 Category 10: Evaluation

This category establishes how robustness conclusions are reassessed over time, how unresolved robustness work is prioritised, and how failures are validated across system layers.

5.10.1 MR-10.1: Continuous Robustness Assessment. This MR means checking again whether an earlier robustness conclusion still holds after the system or its operating conditions change. A conclusion reached before deployment may no longer be valid after changes to the model, data, deployment environment, protective measures, or attack methods considered in the earlier assessment [9, 55, 70]. The audit should therefore require a new assessment at planned intervals and after important changes, such as model updates, retraining, deployment changes, robustness failures, attack methods discovered or reported after the earlier assessment, or changes to measures used to reduce those failures [19, 64, 83, 99, 110, 116]. Each new assessment should identify which earlier robustness conclusions may have been affected and whether they are still supported by the available evidence [22, 61, 73, 76, 82, 125].

5.10.2 MR-10.2: Robustness-aware Progress Tracking and Prioritisation. Progress tracking means keeping a clear record of each robustness problem until it has been resolved or formally accepted. Without such tracking, problems may be forgotten, repeated, or left unresolved [19, 22, 82, 83]. The audit should record the problem, its current status, the action required, and the evidence used to judge whether it has been resolved [19, 22, 83]. Problems should be addressed first when they are likely to cause more serious or repeated consequences, or affect more users [96, 101, 131]. As a follow-up, the audit should test the problem again after corrective action to determine whether it still remains [8, 19, 61].

5.10.3 MR-10.3: Cross-Layer Failure Validation. Cross-layer failure validation means checking whether a robustness problem stems from interactions between different parts of the AI-enabled system rather than the model alone. Such problems may involve the model, software framework, external service, infrastructure, runtime process, or human interaction [131]. The audit should identify the parts involved, reproduce the suspected failure, observe what happens, and determine whether the evidence supports the proposed cause [9, 24, 80, 99, 131]. If some interactions cannot be tested, the audit should state which ones remain unchecked [131].

5.11 Category 11: Monitoring

This category establishes how deployed AI-enabled systems are observed for robustness-relevant change, runtime violations, and auditable operational records after the original assessment.

5.11.1 MR-11.1: Continuous, Automated, and Real-time Robustness Monitoring. Robustness monitoring means observing a deployed AI-enabled system for changes that may reduce its ability to behave reliably. Such changes may include worsening performance, changes in incoming data, unusual outputs, signs that the system is being deliberately manipulated, or changes in operating conditions [8, 9, 19, 20, 39, 55, 70, 78, 80, 86, 89, 101, 116, 120]. Monitoring may be continuous, meaning that the system is checked throughout operation, and automated where software can perform those checks without repeated manual inspection. Where rapid response is required, changes should be detected and reported as they occur or soon afterwards. The audit should state what is monitored, how often it is checked, and what change should trigger further investigation or reassessment (see MR-10.1 and MR-10.2) [8, 15, 19, 53, 61, 75, 91, 101]. Automated alerts should be checked before being treated as evidence that robustness has degraded [4, 9, 76, 86, 99, 106].

5.11.2 MR-11.2: Robustness-aware Logging. This means keeping records that allow auditors to reconstruct what happened when a robustness-related problem occurs. Without such records, it may be difficult to determine what the system received, what it produced, which model version was running, or what changes had been made [70, 78]. The audit should therefore check whether logs capture the information needed to investigate the event, including relevant inputs, outputs, model versions, system settings, alerts, and recorded failures [70, 99, 106, 116]. The logs should be kept for a defined period, protected from unauthorised access or alteration, and available when later investigation is required [70, 106, 116].

5.12 Category 12: Communication

This category establishes how evidence from robustness audits is communicated through visual interfaces, risk reports, and user-facing documentation so the intended audience can inspect, interpret, and act on audit results.

5.12.1 MR-12.1: Robustness Dashboards and Visualisation. This MR entails presenting robustness audit results in a visual form so that users can quickly see important robustness changes, problems, and trends. This is useful when relevant information is spread across different measures, AI-enabled system components, or periods of system operation [70, 89]. The audit should check whether the visualisation clearly shows the robustness measures being tracked, important changes over time, detected problems, and any uncertainty that affects interpretation [4, 78, 84, 99, 101, 117, 129]. It should also identify who the visualisation is intended for and present enough detail for that audience to understand the result without hiding important problems or limitations [1, 60, 89, 90, 101].

5.12.2 MR-12.2: Robustness Documentation and Risk Report. The written audit report provides a clear record of what was assessed, how the assessment was performed, what was found, and what limits the conclusion. Without this information, readers may not know which evidence supports the robustness conclusion or where uncertainty remains [22, 53, 61, 64, 76, 82, 83, 91, 96, 99]. The report should state the audit objective, robustness claim, methods used, evidence examined, main findings, known limitations, unresolved problems, and any required follow-up actions. It should also identify the system, model, data, and test conditions to which the conclusion applies, drawing on retained operational records where needed (see MR-11.2) [8, 9, 21, 39, 54, 61, 69, 90, 93, 94, 110, 118, 120, 131].

5.13 Category 13: Efficiency

This category examines whether robustness auditing activities can be executed within financial, temporal, computational, and operational constraints without overstating what constrained evidence can support.

5.13.1 MR-13.1: Resource Efficiency and Scalability. This MR concerns whether a robustness audit can be carried out with the available time, computing resources, infrastructure, and people with the required expertise as the size or complexity of the system increases. Audits may become difficult to perform when models, datasets, or system environments require substantial computation, specialist expertise, or long execution times [20, 24, 39, 53, 55, 60, 62, 73, 75, 76, 85, 105, 116, 117, 125]. The audit should therefore state the resources and time required and whether those limits reduce the amount of testing that can be performed [51, 82, 105, 117, 127]. If resource limits require fewer tests or smaller-scale evaluation, the robustness conclusion should be limited to what was actually assessed [48, 51, 60, 80, 85, 127].

5.14 Category 14: Adaptability

This category establishes whether robustness audits are adapted to the domain, task, model type, data modality, and deployment setting being assessed.

5.14.1 MR-14.1: Domain-specific Adaptability. This MR means adjusting the robustness audit to the characteristics of the application area in which the AI-enabled system is used. A test that is suitable in one domain may be misleading in another because operating conditions, acceptable performance, and the consequences of failure can differ [30, 43]. The audit should therefore adapt the test conditions, measures, and acceptance criteria to the domain, including the type of data, expected operating conditions, and robustness problems that matter for that application [7, 20, 39, 51, 80, 82, 90, 93, 96, 125]. The robustness conclusion should apply only to the domain and conditions that were actually assessed [14, 30, 51, 55, 67, 82, 84, 93, 102, 117].

5.14.2 MR-14.2: Task and Model Versatility. This MR means checking whether an audit method remains suitable when the AI task, model type, or input data format changes. A method designed for one setting may give misleading results when transferred to another without adjustment [54]. The audit should therefore assess whether the chosen method fits the system’s task, model type, input data, and intended use, and should state any limits on using that method in other settings [7, 54, 69]. For example, a method developed for a classification task, a neural-network model, or a single input type may not work in the same way for other tasks, model types, or systems that use text, images, audio, or combinations of these inputs [7]. Where no well-established assessment method exists, the audit should state that limitation rather than assume that one method is suitable for every model or task [116].

5.15 Category 15: Accessibility

This category concerns whether tools for robustness auditing, explanations, visualisations, documents, and outputs are understandable and usable by their intended audiences [42, 91, 133].

5.15.1 MR-15.1: Accessible and User-Friendly Robustness Tools, Visualisations, and Documents. Accessibility means that tools for robustness auditing can be operated, visualisations can be understood, and documents can be read and interpreted by their intended users. User-friendly means that these materials are easy to navigate and use without unnecessary effort. Different users may have different levels of technical knowledge, language needs, or accessibility requirements [16, 42, 54, 70, 91]. The audit should therefore check whether tool interfaces are easy to use, visualisations

have clear labels and explanations, and documents use language and detail suited to the intended audience [16, 91]. Non-technical users should receive clear summaries, while qualified users should be able to access the technical detail needed to examine the robustness evidence [1, 19, 54, 76, 78].

5.16 Category 16: Configurability

This category establishes whether robustness-assessment tools can be configured to match the scoped system, domain, thresholds, perturbations, attack assumptions, and deployment conditions.

5.16.1 MR-16.1: Configurable Robustness Assessment Tools. Configurability means that the settings of a robustness-assessment tool can be changed to match the system and the robustness condition being tested. Fixed default settings may give misleading results when the audit requires different levels of input or operating-condition change, different acceptance criteria, or different assumptions about how inputs or system conditions may be changed [1, 30, 51, 60, 72, 94, 120, 130]. The audit should therefore check whether the required settings can be changed and should state which settings were used and why. Any change from the default settings should be explained, and the selected settings should still match the conditions covered by the robustness claim [30, 51, 60, 72, 94, 130].

Key Findings of RQ1

Finding 1. The literature review produced 43 meta-requirements organised into 16 categories and three dimensions: Human, Functional, and Quality.

Finding 2. Most literature-derived meta-requirements concern audit activities and evidence (22 Functional), compared with organisational and human aspects (16 Human) and audit-approach qualities (5 Quality).

6 Practitioners' Perspectives on the Meta-Requirements

RQ2 examines practitioners' perspectives on the 43 literature-derived meta-requirements evaluated in the survey. The analysis draws on 5,504 Likert ratings from 128 valid respondents and 586 substantive comments. We report respondent characteristics, quantitative support, and qualitative feedback, including proposed refinements, challenges, and additional concerns. These findings informed the final set of meta-requirements presented in Section 5.

6.1 Respondent Characteristics

After screening, we retained 128 valid responses. Respondent characteristics describe the validation sample's experience and demographic composition, including AI/ML roles, AI robustness assessment or AI robustness auditing experience, robustness-related activities, AI/ML experience, location, age, gender, and education.

Professional Background and Experience. Respondents represented diverse AI/ML-related backgrounds, including machine-learning engineers, data scientists, AI/ML auditors, software engineers, data engineers, and other related roles. Figure 7 shows these roles together with respondents' AI robustness assessment or auditing experience. The two largest experience groups had less than one year (40%) and one to three years (41%) of robustness assessment or auditing experience, respectively, while smaller groups reported longer experience or no prior experience. The sample therefore represents respondents with varying levels of AI/ML and robustness-assessment experience, rather than exclusively experienced AI robustness auditors. The findings should be interpreted in light of this participant composition.

Table 8. Overall practitioner validation results across all MR ratings.

Response group	No. ratings	%
Agree / Strongly agree	4,469	81.20
Neutral	709	12.88
Disagree / Strongly disagree	326	5.92
Total	5,504	100.00

Table 9. Distribution of individual Likert responses across all MR ratings.

Likert response	No. ratings	%
Strongly agree	2,624	47.67
Agree	1,845	33.52
Neutral	709	12.88
Disagree	235	4.27
Strongly disagree	91	1.65
Total	5,504	100.00

Table 10. Support classification of rated meta-requirements.

Support status	No. of MRs
Strongly supported	41
Supported	2
Mixed or unclear	0
Challenged	0
Total rated MRs	43

unqualified acceptance. Instead, the quantitative results establish the overall level of practitioner support, while the qualitative comments explain where clarification, refinement, contextualisation, or extension was needed.

The absence of a quantitatively challenged MR does not mean that practitioners raised no concerns. It means no MR reached the predefined aggregate threshold to be classified as challenged. The 326 disagreement-level ratings and the open-text comments therefore remained important for interpreting conditional acceptance, feasibility concerns, scope concerns, and overlap concerns.

6.3 Meta-Requirement-Level Validation

Table 12 summarises the practitioner ratings for the 43 literature-derived meta-requirements. Overall, support was consistently high: agreement or strong agreement ranged from 71.88% to 88.28% across the individual meta-requirements. Of the 43 meta-requirements, 41 were classified as strongly supported and two as supported; none was quantitatively classified as challenged.

MR-8.3, concerning compatibility with AI-enabled system infrastructure, received the highest agreement (88.28%), followed by MR-5.3 on robustness-focused data quality checking (87.50%) and MR-12.2 on documenting robustness risks, evidence, and conclusions (86.72%). The two meta-requirements classified as supported rather than strongly supported were MR-2.2 on risk-based stakeholder involvement (71.88%) and MR-6.5 on robustness-focused training-time mitigation (73.44%). Despite their comparatively lower ratings, both retained clear majority agreement.

6.3.1 Category-Level Validation. The category-level analysis assessed whether practitioner support extended across the original survey structure. All 16 categories received broad agreement-level support (Table 11).

Table 11. Category-level agreement ranking across the original 16 survey categories.

Rank	Category	Agree (%)	Rank	Category	Agree (%)
1	Workflow Integration	86.20	9	Communication	81.64
2	Configurability	83.59	10	Robustness Definition	81.25
3	Data Auditability	83.12	11	Human-in-the-Loop	81.25
4	Knowledge and Skills	82.81	12	Accessibility	81.25
5	Output Auditability	82.81	13	Efficiency	79.69
6	Monitoring	82.81	14	Standards and Governance	79.49
7	Adaptability	82.81	15	Model Auditability	79.38
8	Evaluation	82.55	16	Engagement	76.37

Practitioner support extended across all 16 survey categories. Qualitative feedback identified needs for clearer wording and sharper boundaries between some meta-requirements. We therefore interpreted category-level ratings alongside the open-text feedback.

6.4 Qualitative Practitioner Validation and Meta-Requirement Refinement

Open-text responses complemented the quantitative ratings by showing how practitioners proposed retaining, refining, challenging, or extending the literature-derived meta-requirements. Comments were coded (open coding) as confirmations, refinements, scope clarifications, implementation concerns, additions, merge/split suggestions, or challenges. During reconciliation, we compared each proposed change with the existing meta-requirements and other comments; we treated proposals already covered as refinements rather than new additions, merges, or splits. The final decisions comprised 162 retentions, 394 refinements, 13 no-change cases, and 15 comments not usable for meta-requirement synthesis. Only two comments retained an *Add* decision, both identifying the same previously uncovered concern.

Retention of the literature-derived meta-requirements. Practitioner comments reinforced the broad quantitative support for the literature-derived meta-requirements by explaining why specific requirements remained important. P003 described governance and lifecycle requirements as "essential and foundational for credible audits," while P011 and P034 emphasised continuous and proactive lifecycle integration. P117 similarly highlighted standardised metrics, terminology, governance, and audit procedures as important for consistent, reproducible, and comparable assessments. Comparable confirmations across data, model, output, monitoring, adaptability, and configurability concerns (e.g., P019, P031, P047, P072, P091, P110, P117) supported retention of the underlying requirements while allowing further refinement of their scope and implementation.

Refinement of existing meta-requirements. Refinement was the dominant qualitative outcome. Practitioners generally accepted the literature-derived requirements but requested clearer boundaries, applicability conditions, evidence expectations, and implementation guidance. A recurring concern was balancing standardisation with contextual flexibility (P006, P010, P012, P016, P019, P040, P096, P121, P123). Respondents supported common metrics, terminology, governance, and reporting expectations, while cautioning against requiring identical tools or methods across contexts. P016 explicitly called for a "balance [between] standardization and flexibility," and P123 argued that standardisation should support comparability without constraining new evaluation methods, while also noting an "absence of actionable, tool-agnostic guidance." These comments helped distinguish common expectations for comparability from justified, context-specific choices of tools, metrics, and evaluation methods.

Practitioners also refined audit scope, access conditions, and assurance claims (P002, P006, P017, P024, P092, P112, P123). P002 called for explicit "out-of-scope" boundaries, while P017 and P112 distinguished white-, grey-, and black-box contexts and argued that audit requirements should reflect available access. P123 likewise identified "unclear access scope" as a practical barrier, and P024 and P092 requested clearer assumptions, thresholds, limitations, and evidence conditions. These comments led to making access, exclusions, assumptions, and limitations explicit and aligning assurance claims with available evidence.

Risk proportionality and lifecycle reassessment formed a second cross-cutting pattern (P005, P016, P019, P024, P047, P048, P072, P077, P081, P092, P106, P108, P121). Practitioners argued that stakeholder involvement, review, testing, monitoring, and reassessment should vary with system risk, deployment context, and material change. P121 cautioned against "over-burdening low risk systems," while P024 proposed risk-tiered requirements. P047 stressed that robustness is valid only for a specific context and time, and P058 described it as a "living process." P077 and P081 further supported change-sensitive reassessment through scheduled, event-driven, and incident-driven triggers. These comments refined continuous assessment into a risk-sensitive, change-triggered process rather than repeated testing after every change.

Stakeholder participation, expertise, independence, and human oversight were also refined (P002, P005, P006, P010, P016, P017, P022, P047, P057, P091, P117, P122, P125). P006 called for documenting stakeholder input and resolving conflicting viewpoints, while P057 distinguished stakeholder identification from risk-based prioritisation. P047 argued that domain expertise should shape the test basis, assumptions, risks, and evidence needs. P002 and P057 further stressed sufficient technical access and independence from development or commercial pressures. For human oversight, P002 warned against "rubber-stamping," while P005, P016, and P091 linked involvement to risk, reviewer expertise, defined intervention conditions, and auditable decisions. These comments clarified who should participate, when their input is required, and how to exercise independence and oversight.

A second refinement concerned the distinction between auditing and mitigation and the evidence needed to support conclusions (P034, P037, P047, P058, P072, P091, P112, P123). Practitioners stressed that the presence of a mitigation does not establish robustness; audits should evaluate its effectiveness, conditions, limitations, and trade-offs. P058 warned against "a conflation of auditing with engineering," while P112 called for assessment of the "actual impact and trade-offs" of mitigations. P034, P072, and P091 further emphasised testing mitigations across failure types, stress conditions, users, and deployment scenarios. Related comments requested clearer thresholds, versioning, reproducibility, reporting, scalable testing, and configuration records (e.g., P024, P047, P072, P091, P112, P123). We incorporated these points as refinements where the underlying requirement already existed.

Challenges and qualifications. No literature-derived meta-requirement was removed; however, some practitioner comments questioned whether specific implementation mechanisms should be universal or mandatory. The clearest qualifications concerned workflow integration and communication. P121 described workflow and MLOps integration as "important but not foundational," while P123 questioned treating mature MLOps integration as a prerequisite for valid auditing. P108 and P125 likewise argued that integration depth should depend on system complexity, deployment context, risk, resources, and MLOps maturity. P077 viewed dashboards as optional, whereas P123 and P113 stressed that dashboards should not replace a documented and authoritative evidence base. These comments challenged implementation-specific mechanisms rather than the underlying needs for integration, communication, monitoring, and evidence retention. Other comments qualified how evidence should be interpreted. P058 supported stakeholder involvement but argued that audit conclusions should remain evidence-driven. P077 also questioned whether accessibility should have the same mandatory status as other meta-requirements, although other respondents emphasised accessible, audience-appropriate communication (e.g., P037, P047, P054, P072, P081, P091, P097, P099). These comments were retained as qualifying evidence and used to introduce proportionality, conditionality, and clearer distinctions between required audit outcomes and optional implementation mechanisms.

Practitioner-identified extension. Several comments initially proposed additional requirements, including reproducibility and re-audit, auditor independence, stakeholder-conflict management, competency assessment, risk-tiered assurance, incident response, evidence sufficiency, configuration governance, and regulatory or reporting controls (e.g., P006, P022, P024, P058, P081, P091, P122, P125). Reconciliation against the literature-derived meta-requirements showed that most clarified or operationalised existing requirements and were therefore incorporated as refinements. One concern remained insufficiently represented: assurance for externally controlled components and dependencies. P079 called for addressing "third-party and supply chain risks," including external models, foundation-model APIs, pretrained components, datasets, libraries, plugins, and cloud services, while noting that restricted visibility limits defensible assurance claims. P102 independently highlighted third-party validation and auditing of foundation models and APIs, with related concerns also raised by P111 and P123 (P079, P102, P111, P123). Because P079 and P102 identified the

same uncovered concern, their convergent evidence supported one additional practitioner-informed meta-requirement, MR-8.4, *External Component and Supply Chain Assurance*.

Merge and split proposals. Practitioners identified overlap or uneven granularity among some literature-derived requirements (P034, P057, P081, P091, P092, P113). P034 noted overlap among MR-1.1, MR-1.2, and MR-1.3, while P057 distinguished stakeholder identification (MR-2.1) from risk-based prioritisation (MR-2.2). P113 similarly observed overlap between MR-6.1 and MR-6.5 but supported separating stress-test outcomes from the rationale and effectiveness of training-time mitigation. P081, supported by P037 and P091, proposed distinguishing output-failure identification from post-training mitigation assessment. Broader comments from P091 and P092 also noted overlap across evaluation, monitoring, workflow integration, documentation, accessibility, adaptability, and configurability. No MR-level merge or split was adopted; these comments were used to sharpen requirement boundaries, while the original 16-category structure was retained.

Integrated effect of practitioner validation. Practitioner validation was primarily refinement-oriented rather than replacement-oriented. The literature-derived concerns were broadly retained, while practitioners clarified their scope, applicability, evidence expectations, and implementation. Challenges mainly targeted universal or implementation-specific interpretations, and proposed merges and splits were addressed through boundary clarification or refinement of existing requirements. The main exception was the gap concerning external components and supply chain dependencies, which resulted in one new meta-requirement. Overall, the 43 literature-derived meta-requirements were refined and extended to a practitioner-informed set of 44 meta-requirements while retaining the 16-category structure.

Key Findings of RQ2

Finding 3. Practitioner ratings showed broad support for the literature-derived meta-requirements: 41 of the 43 meta-requirements were strongly supported and two were supported, with MR-level agreement ranging from 71.88% to 88.28%.

Finding 4. Practitioner feedback primarily refined the existing meta-requirements and identified one additional standalone requirement on external-dependency robustness, resulting in a final set of 44 meta-requirements across 16 categories.

Table 12. Practitioner validation results for the 43 literature-derived meta-requirements ($n = 128$ respondents per meta-requirement).

Cat.	MR	Meta-requirement	<i>n</i>	Mean	SD	Median	Mode	IQR	Agree (%)	Neutral (%)	Disagree (%)	Quantitative status
1	MR-1.1	Should use standardized robustness metrics	128	4.23	1.12	5.0	5	1.0	82.81	6.25	10.94	Strongly supported
1	MR-1.2	Should use shared robustness terminology	128	4.15	1.04	4.0	5	1.0	76.56	16.41	7.03	Strongly supported
1	MR-1.3	Should use standardized robustness tools	128	3.90	0.92	4.0	4	0.0	78.12	12.50	9.38	Strongly supported
1	MR-1.4	Should provide actionable audit guidance	128	4.16	1.00	4.0	5	1.0	75.78	17.19	7.03	Strongly supported
1	MR-1.5	Should integrate robustness into AI lifecycle workflows	128	4.30	0.93	5.0	5	1.0	81.25	14.06	4.69	Strongly supported
1	MR-1.6	Should establish robustness governance protocols	128	4.31	0.97	5.0	5	1.0	83.59	10.16	6.25	Strongly supported
1	MR-1.7	Should define audit access requirements	128	4.22	0.96	4.5	5	1.0	79.69	14.06	6.25	Strongly supported
1	MR-1.8	Should define auditor procedures	128	4.17	0.97	4.0	5	1.0	78.12	14.06	7.81	Strongly supported
2	MR-2.1	Should identify diverse stakeholder perspectives, including those affected by robustness failures based on robustness risk	128	4.10	0.96	4.0	5	1.0	76.56	15.62	7.81	Strongly supported
2	MR-2.2	Should prioritize stakeholder involvement by robustness risk	128	4.00	1.00	4.0	5	2.0	71.88	19.53	8.59	Supported
2	MR-2.3	Should include independent robustness review	128	4.18	0.94	4.0	5	1.0	75.78	20.31	3.91	Strongly supported
2	MR-2.4	Should involve experts early in robustness framing	128	4.27	0.92	5.0	5	1.0	81.25	13.28	5.47	Strongly supported
3	MR-3.1	Should scope robustness concerns and audit claims	128	4.27	0.90	5.0	5	1.0	81.25	14.06	4.69	Strongly supported
3	MR-3.2	Should identify assumptions about robustness trade-offs	128	4.26	0.97	5.0	5	1.0	81.25	11.72	7.03	Strongly supported

Continued on next page

Table 12 - continued from previous page

Cat.	MR	Meta-requirement	<i>n</i>	Mean	SD	Median	Mode	IQR	Agree (%)	Neutral (%)	Disagree (%)	Quantitative status
4	MR-4.1	Should involve relevant multidisciplinary robustness expertise	128	4.32	0.96	5.0	5	1.0	81.25	14.06	4.69	Strongly supported
4	MR-4.2	Should provide role-specific robustness training	128	4.24	0.86	4.0	5	1.0	84.38	10.94	4.69	Strongly supported
5	MR-5.1	Should ensure robustness assessment data coverage	128	4.33	0.89	5.0	5	1.0	85.94	9.38	4.69	Strongly supported
5	MR-5.2	Should document data description, provenance, and lineage	128	4.23	0.88	4.0	5	1.0	81.25	13.28	5.47	Strongly supported
5	MR-5.3	Should check data quality for robustness assessment	128	4.48	0.89	5.0	5	1.0	87.50	7.81	4.69	Strongly supported
5	MR-5.4	Should identify subgroups and data-level bias sources	128	4.20	0.89	4.0	5	1.0	83.59	12.50	3.91	Strongly supported
5	MR-5.5	Should apply robustness-focused data preprocessing mitigations	128	4.02	0.96	4.0	4	1.0	77.34	14.84	7.81	Strongly supported
6	MR-6.1	Should perform model robustness stress testing	128	4.29	0.99	5.0	5	1.0	81.25	12.50	6.25	Strongly supported
6	MR-6.2	Should specify model intended use and limits	128	4.16	0.99	4.0	5	1.0	77.34	17.97	4.69	Strongly supported
6	MR-6.3	Should support explainability for robustness assessment	128	4.12	0.89	4.0	4	1.0	80.47	13.28	6.25	Strongly supported
6	MR-6.4	Should identify model-level robustness failure sources	128	4.31	0.99	5.0	5	1.0	84.38	7.03	8.59	Strongly supported
6	MR-6.5	Should apply robustness-focused training-time mitigations	128	3.94	0.94	4.0	4	2.0	73.44	21.09	5.47	Supported
7	MR-7.1	Should identify output-level bias or failure sources and assess whether post-training mitigations effectively address them	128	4.27	0.95	5.0	5	1.0	82.81	11.72	5.47	Strongly supported
8	MR-8.1	Should integrate robustness checks into AI development workflows	128	4.34	0.85	5.0	5	1.0	85.94	10.16	3.91	Strongly supported
8	MR-8.2	Should embed robustness checks in ML/MLOps pipelines	128	4.27	0.84	4.0	5	1.0	84.38	13.28	2.34	Strongly supported
8	MR-8.3	Should ensure compatibility with AI-enabled system infrastructure	128	4.27	0.77	4.0	4	1.0	88.28	7.81	3.91	Strongly supported
9	MR-9.1	Should define human participation levels in robustness decisions	128	4.18	0.96	4.0	5	1.0	81.25	12.50	6.25	Strongly supported
10	MR-10.1	Should reassess robustness continuously or when triggered	128	4.25	1.01	5.0	5	1.0	83.59	8.59	7.81	Strongly supported
10	MR-10.2	Should track and prioritize robustness assessment progress	128	4.20	0.88	4.0	5	1.0	79.69	16.41	3.91	Strongly supported
10	MR-10.3	Should validate failures across relevant system layers	128	4.27	0.86	4.0	5	1.0	84.38	12.50	3.12	Strongly supported
11	MR-11.1	Should maintain continuous, automated robustness monitoring	128	4.22	0.93	4.0	5	1.0	81.25	13.28	5.47	Strongly supported
11	MR-11.2	Should log evidence needed to diagnose robustness failures	128	4.33	0.85	5.0	5	1.0	84.38	10.94	4.69	Strongly supported
12	MR-12.1	Should provide robustness dashboards and visualizations	128	3.99	1.04	4.0	4	1.0	76.56	14.06	9.38	Strongly supported
12	MR-12.2	Should document robustness risks, evidence, and conclusions	128	4.38	0.87	5.0	5	1.0	86.72	7.03	6.25	Strongly supported
13	MR-13.1	Should ensure robustness auditing is cost-effective, time-efficient, and scalable	128	4.18	0.93	4.0	5	1.0	79.69	14.84	5.47	Strongly supported
14	MR-14.1	Should adapt robustness assessment to the deployment domain	128	4.26	0.85	4.0	5	1.0	85.94	10.16	3.91	Strongly supported
14	MR-14.2	Should support relevant tasks, models, and modalities	128	4.18	0.90	4.0	5	1.0	79.69	14.06	6.25	Strongly supported
15	MR-15.1	Should make robustness tools, visualizations, and related outputs accessible to intended users	128	4.21	0.99	4.0	5	1.0	81.25	11.72	7.03	Strongly supported
16	MR-16.1	Should allow configuration of robustness assessment tools	128	4.20	0.87	4.0	5	1.0	83.59	10.94	5.47	Strongly supported

Note. Agree (%) combines *Agree* and *Strongly agree*; Disagree (%) combines *Disagree* and *Strongly disagree*.

7 Discussion

This section interprets the main insights arising from the literature synthesis and practitioner validation, relates them to prior work, and considers their implications for AI robustness auditing research and practice.

7.1 A Robustness Test Is Necessary but Not Sufficient for a Robustness Audit

The derived meta-requirements reinforce the distinction between robustness testing and auditing: tests provide behavioural evidence, whereas an audit evaluates whether that evidence supports a defined robustness claim within specified scope, criteria, assumptions, and limitations (see Sections 5.3 and 5.6). Prior work reflects this division: robustness research develops benchmarks, metrics, and evaluation methods for generating such evidence [30, 46, 115], whereas AI auditing research addresses how evidence is scoped, documented, and interpreted for assurance decisions [81, 82, 99]. Practitioner feedback reinforces this interpretation; P123 emphasised documented evidence, limitations, and reproducibility conditions (see Section 6.4). Thus, a successful robustness test supports an audit conclusion only when its conditions and evidence are traceable to the robustness claim being assessed.

7.2 From Model-Level Robustness to System-Level Assurance

Our findings indicate that model-level robustness cannot, by itself, establish the robustness of the deployed AI-enabled system, because failures may arise from data, preprocessing, software components, output controls, or deployment workflows (see Sections 5.5-5.8). Model-level evidence must therefore be interpreted within the wider system boundary and the interactions that can affect the robustness claim. This interpretation aligns with prior work distinguishing

model- from system-level assessment [82, 89] and showing that failures can originate outside the model and propagate across components [131]. Practitioner feedback supports this view. For example, P031 noted that failures are "rarely isolated to the model" and should be assessed across the system (see Section 6.4). Accordingly, robustness auditing should treat model-level evidence as one component of system-level assurance and bound audit conclusions to the deployed system and relevant component interactions.

7.3 Strong Literature-Practice Convergence with Contextual Qualifications

The RQ2 findings indicate substantial convergence between the literature-derived meta-requirements and practitioner judgement, but not unconditional acceptance. Practitioner feedback largely retained the underlying concerns while refining their scope, access assumptions, feasibility, and contextual applicability (see Section 6.4). P016's call to balance "standardization with flexibility" captures this qualification. Prior work similarly suggests that common assurance requirements can provide a shared baseline, while their operationalisation depends on system and deployment context [22, 30, 116]. The implication is therefore not a fixed audit procedure, but a stable set of robustness auditing concerns whose implementation should remain context-sensitive. Within the studied practitioner sample, literature provides a useful foundation for *what* should be addressed, while practitioner input helps determine *how* those requirements should be applied.

7.4 Robustness Is Conditional, Not Universal

Our findings indicate that robustness depends on the claim being assessed rather than being a context-free property of an AI-enabled system. The relevant system boundary, protected behaviour, operating conditions, disturbances or threats, and acceptance criteria determine what the available evidence can support (see Sections 5.3 and 5.14). Practitioner feedback reinforces this interpretation: P002 called for explicit "out-of-scope" boundaries, while other respondents emphasised context-appropriate threat models, thresholds, operating conditions, and evaluation methods (see Section 6.4). Prior work similarly treats robustness as dependent on assessment conditions and context-specific assurance criteria [30, 82, 116]. Accordingly, audit conclusions should state what is claimed to be robust, under which conditions, and against which criteria, rather than generalising beyond the assessed context.

7.5 Robustness Assurance Is Time-Bounded and Requires Reassessment

Our findings indicate that robustness assurance remains valid only while the conditions supporting the original conclusion remain applicable. Changes to models, data, deployment conditions, threats, or other relevant factors can therefore weaken earlier evidence (see Sections 5.10 and 5.11). Prior work similarly treats AI assurance as an ongoing activity in which material system changes may require renewed evaluation [73, 82, 110, 116]. Practitioner feedback adds an important qualification: P047 described robustness as valid only for a specific context and time, while P077 cautioned against reassessment after every minor change (see Section 6.4). Reassessment should therefore be triggered by changes that can materially affect the robustness claim, with its scope proportionate to their significance and risk. Robustness assurance is thus better treated as a maintained conclusion than as a one-time judgement.

7.6 An Auditable AI-Enabled System Requires an Audit-Ready Organisation

Our findings indicate that technical auditability alone is insufficient for credible robustness auditing. Governance, engagement, competence, and human-oversight requirements show that robustness evidence must be interpreted and acted on by people with appropriate expertise, independence, authority, and defined responsibilities (see Sections 5.1,

5.2, 5.4, and 5.9). Prior work similarly frames AI auditing as an organisational and interdisciplinary activity rather than a purely technical assessment [16, 22, 70, 99]. Practitioner feedback reinforces this interpretation: P003 emphasised early expert involvement and independence, while P006 highlighted documented stakeholder input and resolution of conflicting viewpoints (see Section 6.4). Accordingly, audit readiness should cover both the AI-enabled system and the organisation by defining required expertise, independence, responsibilities, decision authority, and procedures for handling disagreements and follow-up actions.

7.7 External Components Extend the Scope of Robustness Audits

Our findings indicate that robustness assurance may depend on components outside the audited organisation’s direct control. External models, APIs, datasets, libraries, plugins, and cloud services can affect the robustness claim while limiting auditor visibility and access to evidence. Practitioner feedback exposed this gap: P079 highlighted “third-party and supply chain risks,” while P102 identified foundation models and third-party APIs as requiring explicit audit attention (see Section 6.4). Prior work similarly shows that assurance evidence may be distributed across providers, deployers, and auditors [64, 82, 116]. MR-8.4 therefore extends the audit boundary to relevant external dependencies by requiring the audit to identify them, assess how their failure or change could affect the robustness claim, and state how limited access constrains the robustness conclusion [9, 14, 24, 75, 82, 131]. Relevant records for identifying and tracing these external dependencies may include SBOMs and AIBOMs [88, 98]. Because MR-8.4 emerged after the original requirements were rated, its practical value requires further empirical evaluation.

7.8 Implications for Practice

The meta-requirements are intended to support the completeness and design of AI robustness audits by specifying what should be defined, tested, documented, assigned, monitored, and reassessed. They may also complement instruments such as the EU AI Act and NIST AI RMF [70, 116], although no systematic mapping to their provisions was performed.

7.8.1 For AI Robustness Auditors. Our findings imply that robustness audits should begin with a clearly defined robustness claim rather than an available test. Auditors should establish the system boundary, operating conditions, relevant threats or disturbances, access conditions, metrics, and decision criteria before selecting evaluation methods [82, 99]. For example, these requirements may translate into distribution-shift tests and reassessment triggers for an image classifier, or jailbreak conditions, tool interactions, and dependency evidence for an LLM-based system. Accordingly, audit findings should remain traceable to the supporting evidence and evaluated conditions, while limitations in evidence or access should be stated explicitly (see Sections 7.1 and 7.4).

7.8.2 For Providers and Deployers of AI-Enabled Systems. Our findings indicate that providers and deployers should maintain audit-ready evidence throughout the AI-enabled system lifecycle. This includes preserving relevant system versions, data provenance, configurations, evaluation settings, monitoring records, incidents, external dependencies, and robustness decisions [70, 99, 110]. Accordingly, changes that may affect a robustness claim should be traceable to the reassessment undertaken and any resulting follow-up actions (see Sections 7.2 and 7.5).

7.8.3 For Builders of Robustness Auditing Tools. Our findings indicate that robustness testing tools are insufficient on their own for AI robustness auditing. Existing toolkits can generate adversarial examples, run perturbation tests, measure robustness, or identify vulnerabilities [1]. However, tools that support robustness auditing should also link the robustness claim and scope to the test configuration, system versions, supporting evidence, findings, mitigation status,

retesting, and reassessment [99]. Accordingly, different user views should remain linked to the same audit evidence record to support consistent, traceable reporting.

7.8.4 For Standards and Certification Bodies. Our findings suggest that standards and certification schemes should define common requirements for substantiating robustness claims rather than mandate a single test, metric, or threshold. These requirements should cover terminology, claim definition, minimum evidence, access disclosure, auditor competence, reporting, reassessment, and closure of audit findings [89, 116]. However, assessment methods and thresholds should remain adaptable to the domain, system type, risk level, and deployment context (see Section 7.4). Accordingly, standards should require such choices and deviations to be justified and should state how restricted access or external dependencies limit the assurance that can be claimed.

7.9 Implications for Research on Robustness Auditing

The findings also identify several directions for robustness auditing research. In particular, practitioner validation establishes the relevance and interpretation of the meta-requirements, but further work is needed to determine how they should be operationalised and whether their use improves audit practice.

7.9.1 Operationalising the Meta-Requirements through a Robustness Auditing Artefact. This study identifies and validates meta-requirements for AI robustness auditing but does not specify how to implement them. Future design-science research should therefore translate them into a concrete audit artefact that defines procedures, links robustness claims to evidence, specifies access and decision rules, assigns responsibilities, and supports reassessment. Building on structured AI auditing processes [82, 99], such an artefact should be evaluated for usability, evidence completeness, traceability, and integration with existing development, MLOps, governance, and assurance processes. This would establish how the meta-requirements can be applied consistently while allowing adaptation to different AI-enabled systems and audit contexts.

7.9.2 Evaluating the Meta-Requirements in Real Audit Engagements. Practitioner validation establishes perceived relevance, but it does not show that applying the meta-requirements improves the quality of an actual robustness audit. Future case studies and action-research engagements should therefore apply the meta-requirements to AI-enabled systems that vary in domain, architecture, risk, lifecycle stage, and audit-access level. Such studies could examine audit coverage, evidence completeness, traceability, effort, finding quality, follow-up actions, and decision usefulness. This would extend prior work calling for structured AI auditing processes to be evaluated within organisational practice [82, 99]. The objective is to determine which meta-requirements are practically effective, which remain difficult to operationalise, and where contextual adaptation is necessary without weakening the evidential basis of the audit.

7.9.3 Assessing Inter-Auditor Consistency and Repeatability. A further open question is whether the meta-requirements help different auditors reach sufficiently consistent and explainable conclusions. The present validation examined practitioner support for the meta-requirements, not their consistency of application. Future studies could therefore ask multiple auditors to assess the same AI-enabled system and compare the scopes they define, the evidence they request, the tests they select, the findings they identify, and the conclusions they reach. Where feasible, comparison with audits conducted without guidance from the meta-requirements could test whether their use improves traceability, transparency, and repeatability. The goal should not be identical audit reports, since professional judgement may legitimately differ across contexts, but rather to determine whether the meta-requirements makes such differences explicit and defensible through documented scope, assumptions, evidence, access limitations, and decision criteria.

8 Threats to Validity

We considered potential threats to external, construct, internal, and conclusion validity [128] arising from the literature synthesis and practitioner validation.

8.1 External Validity

External validity concerns the extent to which the findings generalise beyond the reviewed literature and practitioner sample [128]. Relevant studies may have been missed because AI robustness auditing spans multiple research communities and terminology, and relevant evidence may exist outside academic literature. In particular, the practitioner-oriented and interview or survey search terms may have underrepresented technically relevant robustness studies that did not use such terminology. We mitigated this through multiple database searches, pilot-refined search strings, supplementary Google Scholar screening, and explicit eligibility criteria (see Section 4.2). Nevertheless, the 74 retained studies represent a systematically selected academic evidence base rather than an exhaustive account; proprietary, unpublished, non-English, and other non-academic practices may be underrepresented. We recruited participants for the main survey through GitHub and LinkedIn using purposive non-probability sampling (see Section 4.4.3), introducing possible self-selection and coverage bias. Although we targeted participants with relevant AI/ML, software engineering, robustness, governance, monitoring, and auditing backgrounds and collected profile information to characterise the sample, it is not statistically representative of the wider practitioner population. The findings may therefore transfer differently across stakeholder groups, domains, system types, and organisational settings.

8.2 Construct Validity

Construct validity concerns whether the study adequately captures the concepts it investigates [128]. Concepts such as robustness, auditing, monitoring, and assurance may be interpreted differently across research and professional communities, and the technology-agnostic formulation of the meta-requirements may have led respondents to interpret their operational meaning differently. Moreover, a five-point Likert rating captures agreement but does not establish shared interpretation, implementation feasibility, or practical effectiveness. We mitigated these risks by grounding the meta-requirements in source-traceable literature evidence (see Section 4.3), organising them into explicit audit categories, and piloting the survey for terminology, wording, presentation, and completion effort (see Section 4.4). Likert ratings were complemented by open-text feedback on ambiguity, overlap, feasibility, and missing issues. Nevertheless, some interpretive variation remains across heterogeneous professional and system contexts.

8.3 Internal Validity

Internal validity concerns potential bias introduced by the research process [128]. Study screening and the derivation of meta-requirements required researcher judgement, particularly for borderline studies, overlapping concepts, and abstraction decisions. We mitigated this through explicit eligibility criteria, independent screening by two researchers, Cohen’s κ assessment before reconciliation ($\kappa = 0.882$ for title/abstract and $\kappa = 0.741$ for full text), consensus resolution of screening disagreements, source-to-requirement traceability, and review by a second researcher of extracted patterns, category assignments, and meta-requirement formulations, with remaining ambiguities resolved through discussion (see Sections 4.2 and 4.3). Nevertheless, other researchers may make different decisions for some studies or formulate particular meta-requirements differently. The practitioner phase may also be affected by response and interpretation bias. Rating 43 meta-requirements could cause fatigue or satisficing, while the desirable nature of robustness, governance, and

auditability may encourage favourable responses. Pilot testing addressed clarity and completion effort, and open-text feedback on qualifications, implementation concerns, and challenges complemented the agreement ratings. Qualitative coding still involved researcher interpretation; despite structured coding and traceable coding and synthesis decisions (see Section 4.4.6), some comments may reasonably be interpreted or mapped differently.

8.4 Conclusion Validity

Conclusion validity concerns whether the conclusions are supported by the analysed evidence [128]. High practitioner agreement supports the literature-derived meta-requirements but does not establish that applying the meta-requirements improves audit quality, robustness outcomes, or inter-auditor consistency. Likewise, open-text comments capture how respondents interpreted and refined the meta-requirements, rather than prevalence in the wider practitioner population. We therefore analysed the ratings descriptively and integrated them with qualitative evidence rather than treating quantitative agreement as sufficient validation. We compared candidate additions against the existing meta-requirements to avoid isolated or overlapping requirements. This synthesis produced one new requirement, MR-8.4, which was not quantitatively rated because it emerged after the original 43 meta-requirements had been evaluated. Thus, the study provides content-oriented practitioner validation rather than evidence of operational effectiveness; further evaluation through real audit engagements and inter-auditor studies is required.

9 Conclusion and Future Work

AI robustness auditing knowledge is distributed across multiple research domains, including AI, software engineering, and auditing. This study addressed this gap by deriving literature-grounded meta-requirements for AI robustness auditing and evaluating them with practitioners. The systematic literature review retained 74 studies and produced 43 literature-derived meta-requirements across 16 categories. The findings show that robustness auditing extends beyond model-level testing to scoping robustness claims, assessing available evidence and system components, documenting limitations, and continuing monitoring and reassessment. The practitioner survey with 128 participants showed broad support, with 81.20% of ratings indicating agreement or strong agreement and 41 of the 43 literature-derived requirements classified as strongly supported. Practitioner feedback nevertheless refined requirement definitions, scope, implementation conditions, stakeholder responsibilities, workflow integration, monitoring, and reporting. Integrating both evidence sources resulted in a final practitioner-informed set of 44 meta-requirements across 16 categories, including one additional requirement addressing external components and supply chain dependencies. The final set of meta-requirements organises the identified auditing concerns into 44 meta-requirements covering areas such as audit scope and responsibilities, data and model assessment, evidence, workflow integration, monitoring, reporting, and reassessment. Future work will operationalise the meta-requirements into a decision model for selecting appropriate requirements, evidence, tests, metrics, and assessment conditions for a given AI-enabled system and audit context, and subsequently integrate this model into a design support system for practical AI robustness auditing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. This research received funding from Deloitte, as disclosed in the Acknowledgments. Deloitte had no role in the study design, data collection, analysis, interpretation of results, or decision to submit the manuscript for publication.

Acknowledgments

This research has been supported by the Safeguard project (grant No. 20506020). It has also received funding from Deloitte. The authors thank all practitioners who participated in the validation survey and provided feedback on the proposed AI robustness auditing meta-requirements. We also thank colleagues and professional networks who helped distribute the pilot survey, as well as those who provided constructive comments on earlier versions of the manuscript.

Ethics Approval

The practitioner survey received ethical clearance from the Research Ethics and Data Management Committee of the Tilburg School of Humanities and Digital Sciences, Tilburg University (REDC 2026.04). We obtained informed consent from all survey participants before participation.

Data Availability Statement

For transparency and reproducibility, we provide a replication package¹ containing the review protocol, search strings, included-study and quality-assessment records, extraction and coding records, meta-requirement tables, survey instrument, practitioner-validation data, and analysis outputs.

References

- [1] Avinash Agarwal and Manisha J Nene. 2025. Advancing trustworthy ai: A comparative evaluation of ai robustness toolboxes. *SN Computer Science* 6, 3 (2025), 234.
- [2] Alan Agresti. 2010. *Analysis of Ordinal Categorical Data* (2 ed.). John Wiley & Sons, Hoboken, NJ.
- [3] Gopika Ajaykumar, Maureen Steele, and Chien-Ming Huang. 2021. A survey on end-user robot programming. *ACM Computing Surveys (CSUR)* 54, 8 (2021), 1–36.
- [4] Saara Ala-Luopa, Thomas Olsson, Kaisa Väänänen, Maria Hartikainen, and Jouko Makkonen. 2024. Trusting intelligent automation in expert work: Accounting practitioners' experiences and perceptions. *Computer Supported Cooperative Work (CSCW)* 33, 4 (2024), 1343–1371.
- [5] Laith Alzubaidi, Aiman Al-Sabaawi, Jinshuai Bai, Ammar Dukhan, Ahmed H Alkenani, Ahmed Al-Asadi, Haider A Alwzawzy, Mohamed Manoufali, Mohammed A Fadhel, Ahmed Shihab Albahri, et al. 2023. Towards risk-free trustworthy artificial intelligence: Significance and requirements. *International Journal of Intelligent Systems* 2023, 1 (2023), 4459198.
- [6] Anu Bajaj and Om Prakash Sangwan. 2019. A systematic literature review of test case prioritization using genetic algorithms. *Ieee Access* 7 (2019), 126355–126375.
- [7] Alan Balendran, Céline Beji, Florie Bouvier, Ottavio Khalifa, Theodoros Evgeniou, Philippe Ravaud, and Raphaël Porcher. 2025. A scoping review of robustness concepts for machine learning in healthcare. *npj Digital Medicine* 8, 1 (2025), 38.
- [8] Firas Bayram and Bestoun S Ahmed. 2025. Towards trustworthy machine learning in production: An overview of the robustness in mlops approach. *Comput. Surveys* 57, 5 (2025), 1–35.
- [9] Jonathan C. Blood, Nathan W. Herbert, and Martin R. Wayne. 2023. Reliability Assurance for AI Systems. In *2023 Annual Reliability and Maintainability Symposium (RAMS)*. IEEE, Orlando, FL, USA, 1–6. doi:10.1109/RAMS51473.2023.10088197
- [10] Chaima Boufaied, Thanh Nguyen, and Ronnie de Souza Santos. 2026. Practitioner Insights on Fairness Requirements in the AI Development Life Cycle: An Interview Study. *ACM Transactions on Software Engineering and Methodology* (2026). doi:10.1145/3821435
- [11] Jonathan W Browning and John Bustard. 2024. A systematic literature review of entrepreneurial education in electrical, electronic, and computer engineering curricula. *IEEE access* 12 (2024), 7927–7941.
- [12] Bert-Jan Butijn, Damian A. Tamburri, and Willem-Jan van den Heuvel. 2020. Blockchains: a systematic multivocal literature review. *ACM Computing Surveys (CSUR)* 53, 3 (2020), 1–37. doi:10.1145/3369052
- [13] Edna Dias Canedo, Ian Nery Bandeira, Angelica Toffano Seidel Calazans, Pedro Henrique Teixeira Costa, Emille Catarine Rodrigues Cançado, and Rodrigo Bonifácio. 2023. Privacy requirements elicitation: a systematic literature review and perception analysis of IT practitioners. *Requirements Engineering* 28, 2 (2023), 177–194.
- [14] Stephen Casper, Carson Ezell, Charlotte Siegmans, Noam Kolt, Taylor Lynn Curtis, Benjamin Bucknall, Andreas Haupt, Kevin Wei, Jérémy Scheurer, Marius Hobbhahn, et al. 2024. Black-Box Access is Insufficient for Rigorous AI Audits. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, New York, NY, USA, 2254–2272. doi:10.1145/3630106.3659037

¹<https://doi.org/10.5281/zenodo.23267777>

- [15] Vinay Chamola, Vikas Hassija, A Razia Sulthana, Debshishu Ghosh, Divyansh Dhingra, and Biplab Sikdar. 2023. A review of trustworthy and explainable artificial intelligence (XAI). *IEEe Access* 11 (2023), 78994–79015.
- [16] Bhanu Chander, Chinju John, Lekha Warriar, and Kumaravelan Gopalakrishnan. 2025. Toward trustworthy artificial intelligence (TAI) in the context of explainability and robustness. *Comput. Surveys* 57, 6 (2025), 1–49.
- [17] Kathy Charmaz. 2006. *Constructing grounded theory: A practical guide through qualitative analysis*. sage.
- [18] Lianping Chen, Muhammad Ali Babar, and He Zhang. 2010. Towards an evidence-based understanding of electronic data sources. In *14th Int. Conf. on Evaluation and Assessment in Software Engineering (EASE)*. BCS Learning & Development, Limerick, Ireland, 1–4. doi:10.14236/ewic/EASE2010.17
- [19] Pin-Yu Chen and Payel Das. 2023. AI maintenance: a robustness perspective. *Computer* 56, 2 (2023), 48–56.
- [20] Qi Chen, Liqiang Tan, Jingjing Tang, and Xilong Qu. 2025. AI-Enabled IoT Security: A Survey on Advances, Challenges, and Cross-Domain Collaborative Frameworks. In *Proceedings of the 2025 8th Int. Conf. on Computer Information Science and Artificial Intelligence*. Association for Computing Machinery, New York, NY, USA, 1615–1621. doi:10.1145/3773365.3773619
- [21] A. Feder Cooper, Emanuel Moss, Benjamin Laufer, and Helen Nissenbaum. 2022. Accountability in an Algorithmic Society: Relationality, Responsibility, and Robustness in Machine Learning. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, New York, NY, USA, 864–876. doi:10.1145/3531146.3533150
- [22] Sasha Costanza-Chock, Inioluwa Deborah Raji, and Joy Buolamwini. 2022. Who Audits the Auditors? Recommendations from a Field Scan of the Algorithmic Auditing Ecosystem. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, New York, NY, USA, 1571–1583. doi:10.1145/3531146.3533213
- [23] Manuel De Stefano, Fabiano Pecorelli, Dario Di Nucci, Fabio Palomba, and Andrea De Lucia. 2022. Software engineering for quantum programming: How far are we? *Journal of Systems and Software* 190 (2022), 111326.
- [24] Jayati Dev, Nuray Baltaci Akhuseyinoglu, Golam Kayas, Bahman Rashidi, and Vaibhav Garg. 2025. Building guardrails in ai systems with threat modeling. *Digital Government: Research and Practice* 6, 1 (2025), 1–18.
- [25] Ivan R Diamond, Robert C Grant, Brian M Feldman, Paul B Pencharz, Simon C Ling, Aileen M Moore, and Paul W Wales. 2014. Defining consensus: a systematic review recommends methodologic criteria for reporting of Delphi studies. *Journal of clinical epidemiology* 67, 4 (2014), 401–409.
- [26] Natalia Díaz-Rodríguez, Javier Del Ser, Mark Coeckelbergh, Marcos López De Prado, Enrique Herrera-Viedma, and Francisco Herrera. 2023. Connecting the dots in trustworthy Artificial Intelligence: From AI principles, ethics, and key requirements to responsible AI systems and regulation. *Information Fusion* 99 (2023), 101896.
- [27] Julia Dressel and Hany Farid. 2018. The accuracy, fairness, and limits of predicting recidivism. *Science advances* 4, 1 (2018), eaao5580.
- [28] European Parliament and Council of the European Union. 2024. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 Laying Down Harmonised Rules on Artificial Intelligence and Amending Regulations (EC). Official Journal of the European Union, L 2024/1689. <http://data.europa.eu/eli/reg/2024/1689/oj> Published 12 July 2024.
- [29] Mahdi Fahmideh, Aakash Ahmad, Ali Behnaz, John Grundy, and Willy Susilo. 2021. Software engineering for internet of things: The practitioners' perspective. *IEEE Transactions on Software Engineering* 48, 8 (2021), 2857–2878.
- [30] Timo Freisleben and Thomas Grote. 2023. Beyond generalization: a theory of robustness in machine learning. *Synthese* 202, 4 (2023), 109.
- [31] Matthew G Gaber, Mohiuddin Ahmed, and Helge Janicke. 2024. Malware detection with artificial intelligence: A systematic literature review. *Comput. Surveys* 56, 6 (2024), 1–33.
- [32] Matthias Galster, Danny Weyns, Dan Tofan, Bartosz Michalik, and Paris Avgeriou. 2013. Variability in software systems—a systematic literature review. *IEEE transactions on software engineering* 40, 3 (2013), 282–306.
- [33] Vahid Garousi and Michael Felderer. 2017. Experience-based guidelines for data extraction in systematic reviews in software engineering. In *Proceedings of the 21st Int. Conf. on Evaluation and Assessment in Software Engineering*. ACM, Ankara, Turkey, 170–179. doi:10.1145/3084226.3084238
- [34] Vahid Garousi, Michael Felderer, and Tuna Hacaloğlu. 2017. Software test maturity assessment and test process improvement: A multivocal literature review. *Information and Software Technology* 85 (2017), 16–42. doi:10.1016/j.infsof.2017.01.001
- [35] Vahid Garousi, Michael Felderer, and Mika V. Mäntylä. 2019. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology* 106 (2019), 101–121. doi:10.1016/j.infsof.2018.09.006
- [36] Yingqiang Ge, Shuchang Liu, Zuohui Fu, Juntao Tan, Zelong Li, Shuyuan Xu, Yunqi Li, Yikun Xian, and Yongfeng Zhang. 2024. A survey on trustworthy recommender systems. *ACM Transactions on Recommender Systems* 3, 2 (2024), 1–68.
- [37] Margarita Geleta, Jiacen Xu, Manikanta Loya, and Junlin Wang. 2023. Maestro: A Gamified Platform for Teaching AI Robustness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. AAAI Press, Washington, DC, USA, 15816–15824. doi:10.1609/aaai.v37i13.26878
- [38] Shirley Gregor and David Jones. 2007. The Anatomy of a Design Theory. *Journal of the Association for Information Systems* 8, 5 (2007), 312–335. doi:10.17705/1jais.00129
- [39] Ashutosh Gupta, Thomas A Henzinger, Konstantin Kueffner, Kaushik Mallik, and David Pape. 2025. Monitoring robustness and individual fairness. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 790–801.
- [40] Adib Habbal, Mohamed Khalif Ali, and Mustafa Ali Abuzaraida. 2024. Artificial Intelligence Trust, risk and security management (AI trism): Frameworks, applications, challenges and future research directions. *Expert Systems with Applications* 240 (2024), 122442.
- [41] Anne Håkansson, Yigit Can Dündar, and Ronald Lee Hartung. 2023. Towards robustness analysis for adaptive artificial intelligence in multi-autonomous agent systems. *Procedia Computer Science* 225 (2023), 4657–4666.

- [42] Leif Hancox-Li. 2020. Robustness in Machine Learning Explanations: Does It Matter?. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, New York, NY, USA, 640–647. doi:10.1145/3351095.3372836
- [43] Amelia F. Hardy, Anka Reuel, Kiana Jafari Meimandi, Lisa Soder, Allie Griffith, Dylan M. Asmar, Sanmi Koyejo, Michael S. Bernstein, and Mykel John Kochenderfer. 2025. More than Marketing? On the Information Value of AI Benchmarks for Practitioners. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*. Association for Computing Machinery, New York, NY, USA, 1032–1047. doi:10.1145/3708359.3712152
- [44] Petra Heck, Martijn Klabbers, and Marko van Eekelen. 2010. A software product certification model. *Software Quality Journal* 18, 1 (2010), 37–55.
- [45] Petra Heck and Andy Zaidman. 2014. A Quality Framework for Agile Requirements: A Practitioner’s Perspective. arXiv:1406.4692 [cs.SE] <https://arxiv.org/abs/1406.4692>
- [46] Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261* (2019).
- [47] A. R. Hevner, S. T. March, J. Park, and S. Ram. 2004. Design science in information systems research. *MIS Quarterly* 28, 1 (2004), 75–106.
- [48] Levin Hornischer. 2026. Robustness and trustworthiness in AI: a no-go result from formal epistemology. *Synthese* 207, 1 (2026), 22.
- [49] Cornelius Ihle, Dennis Trautwein, Moritz Schubotz, Norman Meuschke, and Bela Gipp. 2023. Incentive mechanisms in peer-to-peer networks—a systematic literature review. *Comput. Surveys* 55, 14s (2023), 1–69.
- [50] Chadni Islam, Muhammad Ali Babar, and Surya Nepal. 2019. A multi-vocal review of security orchestration. *Comput. Surveys* 52, 2 (2019), 1–45. doi:10.1145/3305268
- [51] Haseeb Javed, Shaker El-Sappagh, and Tamer Abuhmed. 2024. Robustness in deep learning models for medical diagnostics: security and adversarial challenges towards robust AI applications. *Artificial Intelligence Review* 58, 1 (2024), 12.
- [52] Davinder Kaur, Suleyman Uslu, Kaley J Rittichier, and Arjan Durrezi. 2022. Trustworthy artificial intelligence: a review. *ACM computing surveys (CSUR)* 55, 2 (2022), 1–38.
- [53] Osman Kaya, AFM Shahen Shah, Muhammet Ali Karabulut, Sumeye Nur Karahan, Mustafa Serdar Osmanca, and Nurettin Acir. 2026. Explainable artificial intelligence (XAI): Concepts, applications, challenges, and future perspectives. *IEEE Access* (2026).
- [54] Nils Kemmerzell, Annika Schreiner, Haroon Khalid, Michael Schalk, and Letizia Bordoli. 2025. Towards a better understanding of evaluating trustworthiness in AI systems. *Comput. Surveys* 57, 9 (2025), 1–38.
- [55] Tushar Khinvasara, Abhishek Shankar, and Connor Wong. 2024. Robustness and reliability testing in healthcare using artificial intelligence. *Asian Journal of Research in Computer Science* 17, 7 (2024), 103–118.
- [56] Barbara Kitchenham, Stuart Charters, et al. 2007. Guidelines for performing systematic literature reviews in software engineering.
- [57] Barbara A. Kitchenham and Shari Lawrence Pfleeger. 2002. Principles of Survey Research: Part 2: Designing a Survey. *ACM SIGSOFT Software Engineering Notes* 27, 1 (2002), 18–20.
- [58] Barbara A. Kitchenham and Shari Lawrence Pfleeger. 2002. Principles of Survey Research: Part 3: Constructing a Survey Instrument. *ACM SIGSOFT Software Engineering Notes* 27, 2 (2002), 20–24.
- [59] Barbara A. Kitchenham and Shari Lawrence Pfleeger. 2008. Personal Opinion Surveys. In *Guide to Advanced Empirical Software Engineering*, Forrest Shull, Janice Singer, and Dag I. K. Sjøberg (Eds.). Springer, 63–92.
- [60] Julia Kokina, Shay Blanchette, Thomas H Davenport, and Dessislava Pachamanova. 2025. Challenges and opportunities for artificial intelligence in auditing: Evidence from the field. *International Journal of Accounting Information Systems* 56 (2025), 100734.
- [61] Adriano Koshiyama, Emre Kazim, Philip Treleaven, Pete Rai, and Lukasz Szpruch. 2024. Towards algorithm auditing: managing legal, ethical and technological risks of AI, ML and associated algorithms. *Royal Society Open Science* 11, 5 (2024).
- [62] Jieren Kou, Xiuyuan Lu, Aobing Yin, Wuyang Zhang, Xinyi Fang, Youzhang Li, Tengyue Pan, Lijia Zhang, Ge Cheng, Hao Yuan, et al. 2024. Beyond Isolated Fixes: A Comprehensive Survey on Hallucination Mitigation with a Three-Dimensional Taxonomy and Integrative Framework. In *2024 7th International Conference on Universal Village (UV)*. IEEE, 1–60.
- [63] Sampo Kuutti, Richard Bowden, Yaochu Jin, Phil Barber, and Saber Fallah. 2020. A survey of deep learning applications to autonomous vehicle control. *IEEE Transactions on Intelligent Transportation Systems* 22, 2 (2020), 712–733.
- [64] Johann Laux, Sandra Wachter, and Brent Mittelstadt. 2024. Trustworthy artificial intelligence and the European Union AI act: On the conflation of trustworthiness and acceptability of risk. *Regulation & Governance* 18, 1 (2024), 3–32.
- [65] Eungyu Lee, Yongsoo Lee, and Taejin Lee. 2023. Adversarial Attack-Based Robustness Evaluation for Trustworthy AI. *Computer Systems Science & Engineering* 47, 2 (2023).
- [66] Jong Seok Lee, Jan Pries-Heje, and Richard Baskerville. 2011. Theorizing in design science research. In *International conference on design science research in information systems*. Springer, 1–16.
- [67] Samuel Lefcourt, Nathaniel Gordon, Hanting Wong, and Gregory Falco. 2022. Robustness Assurance Quotient: Demonstrating Context Matters for AI Performance and ML Security. In *2022 IEEE International Conference on Assured Autonomy (ICAA)*. IEEE, 21–27.
- [68] Stefan Lessmann, Bart Baesens, Hsin-Vonn Seow, and Lyn C Thomas. 2015. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European journal of operational research* 247, 1 (2015), 124–136.
- [69] Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. 2023. Trustworthy AI: From principles to practices. *Comput. Surveys* 55, 9 (2023), 1–46.
- [70] Yueqi Li and Sanjay Goel. 2025. Artificial intelligence auditability and auditor readiness for auditing artificial intelligence systems. *International Journal of Accounting Information Systems* 56 (2025), 100739.

- [71] Johan Linåker, Sardar Muhammad Sulaman, Rafael Maiani de Mello, and Martin Höst. 2015. *Guidelines for Conducting Surveys in Software Engineering*. Technical Report. Department of Computer Science, Lund University.
- [72] Haochen Liu, Yiqi Wang, Wenqi Fan, Xiaorui Liu, Yaxin Li, Shaili Jain, Yunhao Liu, Anil Jain, and Jiliang Tang. 2022. Trustworthy ai: A computational perspective. *ACM Transactions on Intelligent Systems and Technology* 14, 1 (2022), 1–59.
- [73] Michael Madaio, Lisa Egede, Hariharan Subramonyam, Jennifer Wortman Vaughan, and Hanna Wallach. 2022. Assessing the fairness of AI systems: AI practitioners' processes, challenges, and needs for support. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW1 (2022), 1–26.
- [74] Matti Mäntymäki, Matti Minkkinen, Markus Philipp Zimmer, Teemu Birkstedt, and Mika Viljanen. 2023. Designing an AI governance framework: From research-based premises to meta-requirements. (2023).
- [75] Silverio Martínez-Fernández, Justus Bogner, Xavier Franch, Marc Oriol, Julien Siebert, and Adam Trendowicz. 2022. Software engineering for AI-based systems: a survey. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 31, 2 (2022), 1–59.
- [76] Louise McCormack, Malika Bendechache, Dave Lewis, and Diletta Huyskes. 2025. Trust and transparency in AI: industry voices on data, ethics, and compliance. *AI & SOCIETY* (2025), 1–29.
- [77] Ninareh Mehrabi, Fred Morstatter, Kristina Lerman, and Aram Galstyan. 1908. A survey on bias and fairness in machine learning. arXiv.
- [78] Leila Methnani, Manolis Chiou, Virginia Dignum, and Andreas Theodorou. 2024. Who's in charge here? A survey on trustworthy AI in variable autonomy robotic systems. *ACM computing surveys* 56, 7 (2024), 1–32.
- [79] Matti Minkkinen, Joakim Laine, and Matti Mäntymäki. 2022. Continuous auditing of artificial intelligence: a conceptualization and assessment of tools and frameworks. *Digital Society* 1, 3 (2022), 21.
- [80] Sina Mohseni, Haotao Wang, Chaowei Xiao, Zhiding Yu, Zhangyang Wang, and Jay Yadawa. 2022. Taxonomy of machine learning safety: A survey and primer. *Comput. Surveys* 55, 8 (2022), 1–38.
- [81] Jakob Mökander. 2023. Auditing of AI: Legal, ethical and technical approaches. *Digital Society* 2, 3 (2023), 49.
- [82] Jakob Mökander, Jonas Schuett, Hannah Rose Kirk, and Luciano Floridi. 2024. Auditing large language models: a three-layered approach. *AI and Ethics* 4, 4 (2024), 1085–1115.
- [83] Tomas Bueno Momcilovic, Beat Buesser, Giulio Zizzo, Mark Purcell, and Dian Balta. 2026. Assuring EU AI Act Compliance and Adversarial Robustness of LLMs. *Shaping the Digital Future Through Innovation and Practice* (2026), 355.
- [84] Ivana Müller, Katja Dill, and Helena Mihaljević. 2025. Practical Perspectives on Fairness In Technology-Aided Personnel Selection. *Proceedings of the ACM on Human-Computer Interaction* 9, 7 (2025), 1–34.
- [85] Vedant Nanda, Samuel Dooley, Sahil Singla, Soheil Feizi, and John P. Dickerson. 2021. Fairness Through Robustness: Investigating Robustness Disparity in Deep Learning. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, New York, NY, USA, 466–477. doi:10.1145/3442188.3445910
- [86] Sidhant Narula, Mohammad Ghasemigol, Javier Carnerero-Cano, Amanda Minnich, Emil Lupu, and Daniel Takabi. 2025. Exploring Research and Tools in AI Security: A Systematic Mapping Study. *IEEE Access* 13 (2025), 84057–84080. doi:10.1109/ACCESS.2025.3567195
- [87] Roman Nekrasov, Stefano Fossati, Indika Kumara, Damian Andrew Tamburri, and Willem-Jan van den Heuvel. 2026. IaC Generation with LLMs: An Error Taxonomy on Configuration Knowledge Injection. *ACM Transactions on Software Engineering and Methodology* (2026). doi:10.1145/3817608
- [88] Sabato Nocera, Massimiliano Di Penta, Fatima Ahmed, Simone Romano, and Giuseppe Scanniello. 2025. What We Know about AIBOMs: Results from a Multivocal Literature Review on Artificial Intelligence Bill of Materials. *ACM Transactions on Software Engineering and Methodology* (2025).
- [89] Henrik Nolte, Miriam Rateike, and Michèle Finck. 2025. Robustness and Cybersecurity in the EU Artificial Intelligence Act. In *Proc. of the 2025 ACM Conf. on Fairness, Accountability, and Transparency*. ACM, New York, NY, USA, 283–295. doi:10.1145/3715275.3732020
- [90] Luis Oala, Jana Fehr, Luca Gilli, Pradeep Balachandran, Alixandro Werneck Leite, Saul Calderon-Ramirez, Danny Xie Li, Christian Matek, Arun Shroff, Ferath Kherif, Bruno Sanguinetti, and Thomas Wiegand. 2020. ML4H Auditing: From Paper to Practice. In *Proceedings of the Machine Learning for Health NeurIPS Workshop (Proceedings of Machine Learning Research, Vol. 136)*. PMLR, 280–317.
- [91] Victor Ojewale, Ryan Steed, Briana Vecchione, Abeba Birhane, and Inioluwa Deborah Raji. 2025. Towards AI Accountability Infrastructure: Gaps and Opportunities in AI Audit Tooling. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–29. doi:10.1145/3706598.3713301
- [92] Miguel A Olivero, Antonia Bertolino, Francisco José Dominguez-Mayo, Ilaria Matteucci, and María José Escalona. 2022. A delphi study to recognize and assess systems of systems vulnerabilities. *Information and Software Technology* 146 (2022), 106874.
- [93] Tinghui Ouyang, Yoshinao Isobe, Vicent Sanz Marco, Jun Ogata, Yoshiki Seo, and Yutaka Oiwa. 2021. AI robustness analysis with consideration of corner cases. In *2021 IEEE International Conference on Artificial Intelligence Testing (AITest)*. IEEE, 29–36.
- [94] Puntis Palazzolo, Bernd Stahl, and Helena Webb. 2024. Measurable trust: The key to unlocking user confidence in black-box AI. In *Proceedings of the Second International Symposium on Trustworthy Autonomous Systems*. 1–7.
- [95] Cong Pan, Jun You, and Yan Gao. 2023. AI software reliability: concepts and related domains. In *2023 2nd International Conference on Artificial Intelligence and Intelligent Information Processing (AIIIP)*. IEEE, 287–292.
- [96] Eike Petersen, Yannik Potdevin, Philipp Mohammadi, et al. 2022. Responsible and regulatory conform machine learning for medicine: a survey of challenges and solutions. *IEEE access* 10 (2022), 58375–58418.
- [97] Rick Rabiser, Paul Grünbacher, and Deepak Dhungana. 2010. Requirements for product derivation support: Results from a systematic literature review and an expert survey. *Information and Software Technology* 52, 3 (2010), 324–346.

- [98] Petar Radanliev, Omar Santos, Carsten Maple, and Kayvan Atefi. 2026. Operationalising artificial intelligence bills of materials for verifiable AI provenance and lifecycle assurance. *Frontiers in Computer Science* 8 (2026), 1735919.
- [99] Inioluwa Deborah Raji, Andrew Smart, Rebecca N White, Margaret Mitchell, Timmit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. 2020. Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 33–44.
- [100] Pranav Rajpurkar, Emma Chen, Oishi Banerjee, and Eric J Topol. 2022. AI in health and medicine. *Nature medicine* 28, 1 (2022), 31–38.
- [101] Samantha Robertson, Zijie J Wang, Dominik Moritz, Mary Beth Kery, and Fred Hohman. 2023. Angler: Helping machine translation practitioners prioritize model improvements. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–20.
- [102] Pritam Roy. 2024. Enhancing real-world robustness in AI: challenges and solutions. *J. Recent Trends Comput. Sci. Eng* 12, 1 (2024), 34–49.
- [103] Mozamel M Saeed and Mohammed Alsharidah. 2024. Security, privacy, and robustness for trustworthy AI systems: A review. *Computers and Electrical Engineering* 119 (2024), 109643.
- [104] Ana B Sánchez, Pedro Delgado-Pérez, Inmaculada Medina-Bulo, and Sergio Segura. 2022. Mutation testing in the wild: findings from GitHub. *Empirical Software Engineering* 27, 6 (2022), 132.
- [105] Mohd Saqib, Samaneh MahdaviFar, Benjamin CM Fung, and Philippe Charland. 2024. A comprehensive analysis of explainable AI for malware hunting. *Comput. Surveys* 56, 12 (2024), 1–40.
- [106] András Schmelczler and Joost Visser. 2023. Trustworthy and Robust AI Deployment by Design: A framework to inject best practice support into AI deployment pipelines. In *2023 IEEE/ACM 2nd Int. Conf. on AI Engineering-Software Engineering for AI (CAIN)*. IEEE Computer Society, 127–138.
- [107] Muhammad Shafique, Mahum Naseer, Theocharis Theocharides, Christos Kyrkou, Onur Mutlu, Lois Orosa, and Jungwook Choi. 2020. Robust machine learning systems: Challenges, current trends, perspectives, and the road ahead. *IEEE Design & Test* 37, 2 (2020), 30–57.
- [108] Sara Silva and Breno Bernard Nicolau De França. 2023. A Case Study on Data Science Processes in an Academia-Industry Collaboration. In *Proceedings of the XXII Brazilian Symposium on Software Quality*. 1–10.
- [109] Samuel Henrique Silva and Peyman Najafirad. 2020. Opportunities and challenges in deep learning adversarial robustness: A survey. *arXiv preprint arXiv:2007.00753* (2020).
- [110] Katharina Simbeck. 2024. They shall be fair, transparent, and robust: auditing learning analytics systems. *AI and Ethics* 4, 2 (2024), 555–571.
- [111] Nathalie A Smuha. 2025. Regulation 2024/1689 of the Eur. Parl. & Council of June 13, 2024 (eu artificial intelligence act). *International Legal Materials* 64, 5 (2025), 1234–1381.
- [112] Elham Tabassi. 2023. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. Technical Report NIST AI 100-1. National Institute of Standards and Technology, Gaithersburg, MD. doi:10.6028/NIST.AI.100-1
- [113] Mohammad Reza Taesiri, Giang Nguyen, and Anh Nguyen. 2022. Visual correspondence-based explanations improve AI robustness and human-AI team accuracy. *Advances in Neural Information Processing Systems* 35 (2022), 34287–34301.
- [114] Qingzhe Tang, Jingwei Qian, Xiaozhi Du, Shijiao Wang, and Hairui Liu. 2026. Advancing adversarial and LLM robustness in trustworthy AI: a comprehensive survey. *Artificial Intelligence Review* (2026).
- [115] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. 2020. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems* 33 (2020), 18583–18599.
- [116] Eva Thelisson and Himanshu Verma. 2024. Conformity assessment under the EU AI act general approach. *AI and Ethics* 4, 1 (2024), 113–121.
- [117] Andrea Tocchetti, Lorenzo Corti, Agathe Balayn, Mireia Yurrita, Philip Lippmann, Marco Brambilla, and Jie Yang. 2025. AI robustness: a human-centered perspective on technological challenges and opportunities. *Comput. Surveys* 57, 6 (2025), 1–38.
- [118] Ilias Tsingenopoulos, Vera Rimmer, Davy Preuveneers, Fabio Pierazzi, Lorenzo Cavallaro, and Wouter Joosen. 2025. The adaptive arms race: Redefining robustness in ai security. In *2025 28th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*. IEEE, 611–628.
- [119] Ipek Ustunboyacioglu, Indika Kumara, Dario Di Nucci, Damian Andrew Tamburri, and Willem-Jan Van Den Heuvel. 2024. Data Quality Assessment in the Wild: Findings from GitHub. In *Proc. of the 28th Int. Conf. on Evaluation and Assessment in Software Engineering*. 120–129.
- [120] Praneeth Vaddapati. 2026. AI Reliability Gap: Why Large Language Models Fail in Safety-Critical Systems. *Available at SSRN 6349038* (2026).
- [121] Dirk van der Linden, Irit Hadar, and Anna Zamansky. 2019. What practitioners really want: requirements for visual notations in conceptual modeling. *Software & Systems Modeling* 18, 3 (2019), 1813–1831.
- [122] Andreas Vogelsang, Jonas Eckhardt, Daniel Mendez, and Moritz Berger. 2020. Views on quality requirements in academia and practice: commonalities, differences, and context-dependent grey areas. *Information and Software Technology* 121 (2020), 106253.
- [123] Moritz von Zahn, Jan Zacharias, Maximilian Lowin, Johannes Chen, and Oliver Hinz. 2025. Navigating AI conformity: A design framework to assess fairness, explainability, and performance. *Electronic markets* 35, 1 (2025), 24.
- [124] J. G. Walls, G. R. Widmeyer, and O. A. El Sawy. 1992. Building an information system design theory for vigilant EIS. *Information Systems Research* 3, 1 (1992), 36–59.
- [125] Jie Wang, Jun Ai, Minyan Lu, Jieyu Zhao, Haoran Su, Dan Yu, Yutao Zhang, Junda Zhu, and Jingyu Liu. 2026. A Survey of Neural Network Robustness Assessment in Image Recognition. *Comput. Surveys* 58, 12, Article 318 (June 2026), 1–40 pages. doi:10.1145/3814941
- [126] Muhammad Waseem, Peng Liang, Mojtaba Shahin, Amleto Di Salle, and Gastón Márquez. 2021. Design, monitoring, and testing of microservices systems: The practitioners’ perspective. *Journal of Systems and Software* 182 (2021), 111061.
- [127] Wenqi Wei and Ling Liu. 2025. Trustworthy distributed ai systems: Robustness, privacy, and governance. *Comput. Surveys* 57, 6 (2025), 1–42.

- [128] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, Anders Wesslén, et al. 2012. *Experimentation in software engineering*. Vol. 236. Springer, Karlskrona, Sweden.
- [129] Pulei Xiong, Scott Buffett, Shahrear Iqbal, Philippe Lamontagne, Mohammad Mamun, and Heather Molyneaux. 2022. Towards a robust and trustworthy machine learning system development: An engineering perspective. *Journal of Information Security and Applications* 65 (2022), 103121.
- [130] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Russ R Salakhutdinov, and Kamalika Chaudhuri. 2020. A closer look at accuracy vs. robustness. *Advances in neural information processing systems* 33 (2020), 8588–8601.
- [131] Guangba Yu, Gou Tan, Haojia Huang, Zhenyu Zhang, Pengfei Chen, Roberto Natella, Zibin Zheng, and Michael R Lyu. 2026. A survey on failure analysis and fault injection in AI systems. *ACM Transactions on Software Engineering and Methodology* 35, 1 (2026), 1–42.
- [132] Jan Zacharias, Moritz von Zahn, Johannes Chen, and Oliver Hinz. 2022. Designing a feature selection method based on explainable artificial intelligence. *Electronic Markets* 32, 4 (2022), 2159–2184.
- [133] Idrees A Zahid, Salem Garfan, MA Chyad, AS Albahri, OS Albahri, AH Alamoodi, Muhammet Deveci, Raad Z Homod, and Laith Alzubaidi. 2025. Explainability, robustness, and fairness in user-centric intelligent systems: a systematic review. *IEEE Transactions on Emerging Topics in Computational Intelligence* (2025).
- [134] Philipp Zellmer, Jacob Krüger, and Thomas Leich. 2026. Decision making for automotive-platform engineering: A mixed-methods study on practitioners' requirements and criteria. *Journal of Systems and Software* 233 (2026), 112670.