Information and Software Technology 198 (2026) 108232

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Review article

AI-driven self-healing across the edge–cloud continuum: A systematic literature review

Lander Bonilla [a,*], Josu Diaz-de-Arcaya [a], Jon Aguirre-Usandizaga [a], Aitor Almeida [b]

[a] Tecnalia, Basque Research & Technology Alliance (BRTA), C. Leonardo Da Vinci, 11, Miñano, 01510, Álava, Spain
[b] DeustoTech, University of Deusto, Avenida de las Universidades, Bilbao, 48007, Biscay, Spain

ARTICLE INFO

Keywords:
Self-healing
AIOps
Edge–cloud continuum
Systematic literature review
Agentic AI
AI agent
Ethics

ABSTRACT

Context: The increasing complexity posed by Edge–Cloud Continuum architectures requires innovative software development practices and paradigms. Traditional software engineering practices fall short for addressing the particularities of such deployments; hence, there is a critical need for innovative tools and techniques capable of autonomous self-remediation and enhanced resilience with minimal human intervention.
Objective: The overarching goal of this study is to synthesize the state-of-the-art with regard to AI-driven self-healing in the computing continuum. Implementation challenges and ethical implications are analyzed to identify effective approaches, architectures, and future trends in the field of automated program repair.
Methods: A Systematic Literature Review was conducted following the PRISMA 2020 guidelines and Kitchenham's methodology for software engineering. A rigorous automated search across designated academic databases initially yielded 2930 records. After applying strict inclusion and exclusion criteria, 99 primary studies were selected.
Results: The analysis reveals a fundamental paradigm shift from observational monitoring to actionable, autonomous repair. While LLM-driven approaches and Agentic frameworks demonstrate high efficacy in tasks such as causal discovery and complex code patching, a critical validation gap remains regarding their deployment in non-deterministic runtime environments. Current literature often lacks the rigorous testing frameworks required to ensure that AI-generated repairs do not introduce regressions or security vulnerabilities.
Conclusion: Achieving trustworthy autonomy requires a holistic approach that merges advanced AI paradigms with disciplined software engineering. Future research must prioritize the creation of standardized benchmarks and empirical studies to bridge the gap between theoretical algorithms and industrial application. Furthermore, the development of decentralized multi-agent architectures and Human-in-the-Loop techniques is essential to ensure safety, explainability, and ethical compliance in the next generation of resilient software systems.

Contents

* Corresponding author.
E-mail address: lander.bonilla@tecnalia.com (L. Bonilla).

https://doi.org/10.1016/j.infsof.2026.108232
Received 21 January 2026; Received in revised form 5 May 2026; Accepted 12 June 2026
Available online 22 June 2026

1. Introduction

The rapid proliferation of highly distributed and heterogeneous environments has given rise to the Edge–Cloud Continuum, a paradigm that seamlessly integrates diverse data sources with extreme-scale computing resources. However, this evolution has unleashed unprecedented management challenges, leading to a call for 'rebooting' autonomic computing principles to effectively tame the complexity of such interconnected systems [1]. Within this context, the integration of the self-* paradigm, driven by the Monitor, Analyze, Plan, and Execute (MAPE) loop, is no longer an optional optimization but a fundamental requirement to ensure operational efficiency with minimal human intervention [2]. Recent research has emphasized a transition from traditional management models toward autonomic and cognitive architectures, where intelligence is embedded into the system's core to handle the volatility of the computing path [3]. Among these autonomic capabilities, self-healing emerges as a critical frontier for resilience, providing the necessary mechanisms to autonomously detect, diagnose, and repair disruptions. Consequently, autonomous self-healing systems are emerging as a cornerstone of resilience across the cloud–edge continuum, combining AIOps workflows with learning-enabled control loops that can detect, diagnose, and remediate faults with minimal human supervision. The rise of microservices, serverless execution, and highly distributed deployments has amplified operational complexity and the risk of cascading failures, motivating end-to-end automation of incident response [4]. Recent work proposes Agentic, LLM-driven AIOps frameworks that close the loop from observability to action through standardized evaluation environments and fault-injection testbeds, signaling a shift from point algorithms to autonomous orchestration [4,5]. Concurrently, research on root-cause analysis in microservices is moving beyond static correlation toward causal discovery over multivariate time series, improving explainability and localization under temporal dependencies [6,7].

On the repair side, Automated Program Repair (APR) has rapidly advanced with large language models (LLMs): beyond zero/few-shot prompting, comprehensive fine-tuning and Agentic pipelines are demonstrating improved patch quality, while highlighting the need for trustworthy validation beyond test-suite adequacy [8,9]. In parallel, privacy-preserving, distributed learning for anomaly detection is maturing, with federated schemes that sustain accuracy under non-IID data while respecting data governance at the edge [10,11]. These advances, together with explainable anomaly detection methods (e.g., evidential Deep Learning (DL) and correlation-based explanations), reinforce the role of transparency as a prerequisite for safe autonomy in critical systems [12].

Table 1
Research questions and motivations for AI-driven self-healing in distributed environments.

	Questions	Motivation
RQ1	What are the main challenges in integrating AI-driven self-healing into distributed environments?	To comprehend the main challenges in deploying AI-driven self-healing in distributed systems so that professionals can successfully overcome them.
RQ2	What are the most effective AI-based approaches currently used for real-time monitoring of software systems and self-healing?	To identify the most effective AI techniques for real-time monitoring, self-healing, and automated program repair, guiding practitioners toward proven solutions.
RQ3	What architectures, frameworks, and tools support self-healing while enabling the seamless integration of AIOps with MLOps systems?	To have a deeper comprehension of the practical applications of autonomous self-healing in academics and industry.
RQ4	What are the potential risks and ethical implications of using AI-driven techniques for system monitoring and program repair in critical applications?	To examine the risks and ethical challenges of AI-driven monitoring and program repair, ensuring responsible adoption in critical applications.
RQ5	What are the future trends of automated program repair and intelligent monitoring?	To uncover emerging trends in AI-driven self healing, informing future research and development directions.

Combined, the field is converging toward Agentic, explainable, and privacy-aware self-healing: standardized agent benchmarks for AIOps, causality-aware RCA, LLM-centric APR with stronger verification, and federated monitoring across the continuum. This survey positions these threads within a coherent lifecycle and synthesizes recent evidence to guide researchers and practitioners toward robust, scalable, and trustworthy self-healing software.

1.1. Phases of AI-driven self-healing

Modern self-healing systems extend the traditional MAPE-K loop by incorporating Agentic AI and low-latency reflexive actions. The process begins with Monitoring, where multimodal observability is enhanced by edge models and cloud learners. Key innovations include privacy-preserving federated learning and uncertainty-aware detectors that quantify confidence in novel scenarios [10–12], alongside explainable detectors that bolster operator trust [12].

During the Analysis phase, systems utilize temporal and causal structures to localize faults. Neural Granger causal discovery is noted for outperforming static graphical methods, with new benchmarks supporting multi-source root-cause analysis [6,7]. This leads to the Planning stage, where Agentic AIOps frameworks leverage LLMs and standardized interfaces to select mitigation policies [4,5]. Distributed or reflex-augmented loops are increasingly used here to reduce time-to-action [4].

The Execution phase focuses on self-repair, where APR uses fine-tuned LLMs and structured pipelines to generate minimal patches. While validation remains a challenge, these strategies often outperform classical APR tools [8,9]. Underpinning the loop is Knowledge, which manages durable assets like playbooks and causal graphs. Federated optimization ensures these assets are refined continuously without compromising privacy or auditability [10–12].

1.2. Objectives

The overarching goal of this Systematic Literature Review (SLR) is to present a thorough and organized summary of the state of autonomous self-healing systems today. This work dives into the most efficient AI-based methods that enable real-time monitoring, diagnosis, and autonomous recovery; identifies the main organizational and technical obstacles that occur when deploying intelligent self-healing mechanisms across distributed and heterogeneous environments; and examines the architectures, frameworks, and toolchains that enable smooth interoperability between AIOps and self-healing systems. The survey also aims to evaluate the hazards and moral ramifications of using AI-driven monitoring and repair in safety-critical situations. Lastly, the goal of this study is to summarize new developments and prospective research lines that will influence the development of autonomous, reliable, and adaptable self-healing software systems in the future. To achieve this, a list of research questions alongside their objectives are found in Table 1.

The remainder of this paper is structured as follows: The approach used in this SLR is described in Section 2, along with the procedure for retrieving and choosing manuscripts. Section 3 provides a summary of the chosen studies. In order to provide insights into the research questions mentioned in Table 1, Section 4 examines the included studies. The literature's related works and comparable studies are covered in Section 5. Lastly, Section 7 provides a summary of the research findings.

2. Methodology

The methodology of this SLR is grounded in the formal guidelines for Systematic Literature Reviews in Software Engineering proposed by Kitchenham [13,14]. Following this framework, the review process was structured into three main stages: planning, conducting, and reporting. While Kitchenham's guidelines provide the procedural rigor for software engineering research, the PRISMA 2020 statement [15] was employed to ensure transparency and traceability in the reporting of the results.

According to Kitchenham [13,14], article retrieval, search terms, selection criteria, and quality assessment are crucial elements of the Conducting phase. This phase began with defining search terms and a logical strategy for article retrieval across designated databases to mitigate bias and ensure maximal coverage. Next, the identified studies were filtered through a multi-stage selection process using inclusion and exclusion criteria to isolate the relevant primary studies. Finally, the selected research was evaluated using predefined quality metrics to ensure the validity, credibility, and rigor of the evidence used in the synthesis, as recommended for evidence-based software engineering [16].

2.1. Article retrieval

The article retrieval phase constitutes the rigorous and systematic execution of the pre-defined search strategy across designated academic databases and digital libraries. Following the conducting protocol defined by Kitchenham [13,14], this stage necessitates the precise application of the Boolean-logic-derived search string to identify the population of potentially relevant studies. This procedure is critical for mitigating publication bias and ensuring maximal coverage of the literature space, aligning with the evidence-based software engineering principles [13,14]. To maintain full transparency and reproducibility, the entire retrieval process was based on the tool developed in [17]. This automated client streamlines the search by interacting directly with the Scopus API provided by Elsevier [18], a service selected for its superior metadata richness—particularly the availability of abstracts—and its extensive coverage across multiple publishers. While other repositories such as Springer [19], and IEEE [20] were evaluated, the Scopus interface was prioritized. This choice supports the rigorous data management required by Kitchenham's guidelines, as it provides a comprehensive, less error-prone, and unbiased collection of manuscript metadata in a single automated workflow, ensuring that the initial search results are exhaustive and verifiable.

Table 2
Mapping of conceptual categories to search terms used in the SLR.

Category	Concepts (Fig. 1)	Expanded search terms (Keywords)
Application domain	Monitoring & anomaly detection	"system monitoring", "anomaly detection", "outlier detection", "telemetry", "observability"
	Fault diagnosis	"fault diagnosis", "root cause analysis", "fault localization", "troubleshooting", "failure analysis"
	Healing & repairing	"self-healing", "self-repairing", "program repair", "automated recovery", "code correction", "software repair", "bug fixing", "fault correction"
	Learning	"continuous learning", "online learning", "predictive maintenance"
Scope	Challenges	"challenge", "issue", "limitation",
	Frameworks	"framework", "tool", "architecture"
	Ethics	"ethic", "moral", "bias", "fairness", "transparency", "accountability"
	Future trends	"future trend", "future direction", "emerging trend"
AI Paradigms	Generative models	"generative AI", "GAN", "SLM", "diffusion models", "large language model", "LLM", "agentic AI"
	Supervised & Unsupervised models	"deep learning", "CNN", "SVM", "LSTM", "clustering", "K-means", "neural networks"
	Reinforcement learning	"reinforcement learning", "RL", "deep Q-network", "DQN"

Table 3
Inclusion and exclusion criteria applied to the selection process.

ID	Focus	Inclusion	Exclusion
C1	Scope	Articles that address the queries and offer novel ideas.	Articles that do not fit the topic.
C2	Environment	Distributed Continuum.	Articles that do not fit the environment
C3	Contrib. type	Novel algorithms, frameworks, arch., reviews.	Position papers, tutorials, posters, demos.
C4	Doc. type	Peer-reviewed Journals, Conf., Workshops.	Grey lit.; White papers; Non-peer-reviewed; Retracted.
C5	Metadata	Full-text in English; Published 2021–2026.	Non-English; Duplicates; Before 2021; No full-text; Behind paywall

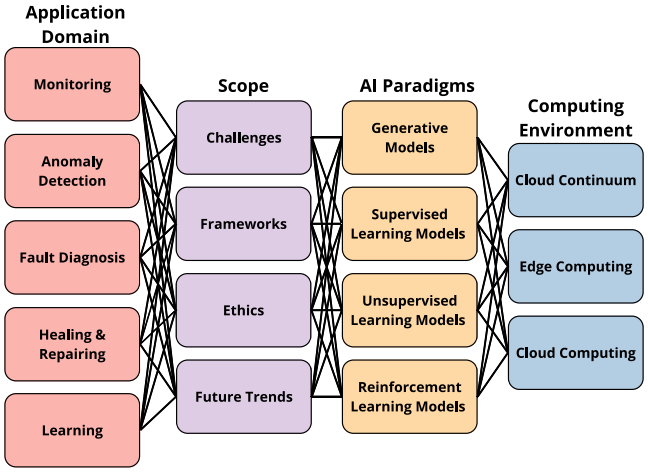

Fig. 1. Search terms used to find the manuscripts that are part of the SLR.

2.2. Search terms

The literature search process was designed to comprehensively identify manuscripts that address the RQs defined within the scope of this study in Table 1. Fig. 1 illustrates the multi-dimensional search taxonomy constructed to identify primary studies for this SLR. The search strategy is grounded in the intersection of four fundamental conceptual pillars: Application Domain, which defines operational objectives such as monitoring, fault diagnosis, and anomaly detection; Scope, which delimits the research focus to specific contributions like frameworks, ethics, and future trends; AI Paradigms, which specifies the methodological approaches including generative, supervised, and reinforcement learning (RL) models; and Computing Environment, which grounds the study within distributed infrastructures like the cloud continuum and edge computing. The reticulated structure of the diagram visually represents the combinatorial logic employed in the database queries, ensuring the selection of manuscripts that address the intersection of these critical dimensions. Table 2 maps the visual concepts from Fig. 1 to specific expanded search terms and keywords used to locate manuscripts for this SLR.

2.3. Selection criteria

Table 3 summarizes the inclusion and exclusion criteria that have been applied in this systematic review to narrow down the number of articles and analyze the most relevant ones for the topic in hand. Considering that AI-driven self-healing is a rapidly evolving concept, and the upsurge of new paradigms such as the Cloud–Edge Continuum, which introduce new challenges in system reliability and orchestration, only studies from 2021 onward have been considered (**C5**). Therefore, the list of articles included must have been identified in the automatic search and hold interesting or groundbreaking ideas where AI is utilized for system management (**C1**) specifically within distributed paradigms (**C2**). To guarantee scientific rigor, the selection is strictly limited to peer-reviewed document types, such as Journals, Conferences, and Workshops (**C4**), that present validated architectures or algorithms (**C3**), while excluding grey literature, short position papers, retracted papers, or demos. On the other hand, articles not written in English are not included. Every effort was made to obtain the articles through available institutional subscriptions; however, as established in **C5**, any manuscript behind a paywall or otherwise unavailable was excluded to ensure the verifiability of the review.

2.4. Quality metrics

The design and reporting of this systematic review were conducted in accordance with the PRISMA 2020 statement [15] and the formal guidelines for Systematic Literature Reviews in Software Engineering proposed by Kitchenham and Brereton [14]. Furthermore, to assess the methodological rigor and scientific impact of the selected studies, a quantitative scoring system was adopted, following the guidelines in [13] and consistent with recent secondary studies in the field of distributed AI [21,22].

Table 4 outlines the specific Quality Metrics defined for this survey. These metrics assess the papers based on content relevance, such as

Table 4
Quality metrics for evaluating the articles.

ID	Quality metric	Score scale
M1	It identifies knowledge gaps to support the study's purpose and offers a thorough state of the art aligned with this SLR.	{0/0, 5/1}
M2	At least one use case aligned with the goals of this SLR is used to validate the research.	{0/0, 5/1}
M3	The quantity of RQs closely related to the paper.	[0...1]
M4	Accessibility via Open Science licenses (Open Access).	{0/1}
M5	Publication type (Journal article, Conference Paper).	[0/0, 5/1]
M6	The manuscript has been published in a top-tier venue.	{0/0, 5/1}
M7	Normalized citation count adjusted by year of publication (freshness-adjusted).	[0...1]

identifying knowledge gaps (**M1**), providing valid use cases (**M2**), and addressing relevant research questions (**M3**). The latter includes accessibility via open science licenses (**M4**), the publication type (**M5**), the prestige of the venue (**M6**), and the article's impact via normalized citation counts (**M7**). The most recent manuscripts are not subject to any restrictions because it is acknowledged that doing so could leave out important content that has not yet been made available to a wider audience. This filter encourages the inclusion of more recent ideas and guarantees the systematic review's freshness.

Each of the listed articles receives a total score between 0 and 10 once these metrics are evaluated. They are categorized as insufficient (0–2), sufficient (3–4), good (5–6), very good (7–8), and excellent (9–10). Please be aware that the mark only approximates the article's alignment with the objectives of this SLR; it has no direct bearing on the article's quality.

3. Results

The purpose of this section is to provide a summary of the chosen studies. The approach used to find and choose the manuscripts for this SLR is provided in Section 3.1. Section 3.2 then explains the validity threats and the potential of bias. Following the auditability principles of Kitchenham [13], Section 3.3 describes the replicability of the filtering process and the availability of the research data. Lastly, a high-level summary of the chosen studies is covered in Section 3.4.

3.1. Study selection

The selection process for this survey was conducted following the PRISMA 2020 guidelines [15], which highlight the necessity of utilizing a flow diagram to explain the outcomes of the search and selection process. The complete flow of information through the different phases of the systematic review is illustrated in Fig. 2.

The initial search across the selected databases yielded a total of 2930 records. This figure represents unique records, as duplicates were automatically filtered during the retrieval process. Prior to the screening phase, 239 records were removed due to technical or format constraints. These exclusions comprised conference reviews ($n = 127$), records with insufficient citations ($n = 107$), errata ($n = 4$), and retracted articles ($n = 1$).

The remaining 2691 records underwent the initial screening process. In this stage, 2319 records were excluded after reviewing their titles ($n = 1537$) and abstracts ($n = 782$), as they did not align with the research topic or the focus defined in the inclusion criteria.

Subsequently, 372 full-text articles were retrieved and assessed for eligibility. These articles were evaluated against the specific inclusion and exclusion criteria detailed in Table 3 (**C1–C5**). Following this assessment, 273 articles were excluded for failing to meet the requirements (e.g., wrong document type, out-of-scope environment, or metadata restrictions). Finally, 99 new studies satisfied all criteria and were included in the final review.

Table 5 presents a detailed breakdown of the manuscripts processed at each stage of the selection workflow, categorized by their respective publishers (e.g., IEEE, ACM, Springer, Elsevier). The Identification phase (**Stages 1** and **2**) consolidates the initial search results and the subsequent removal of technical exclusions such as conference reviews and insufficient citations. During the Screening phase (**Stages 3** and **4**), the authors performed a granular filtering process based first on titles and subsequently on abstracts, which significantly reduced the set of candidate records. Finally, the Eligibility assessment (**Stage 5**) involved a comprehensive full-text review of the remaining articles, resulting in the final set of 99 included studies. This granular view allows for an analysis of the provenance of the most relevant literature in the field. These studies have been thoroughly examined and graded using the metrics specified in Table 4.

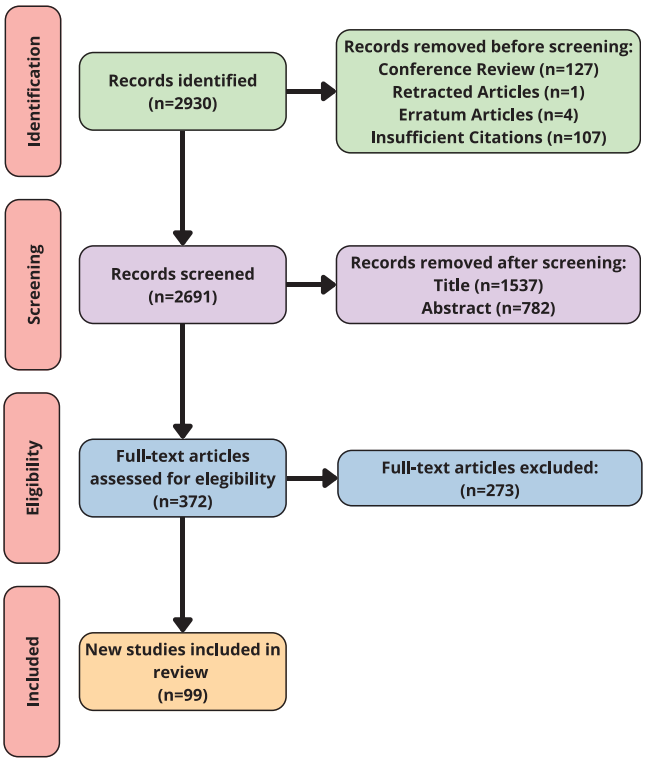

Fig. 2. Flow diagram illustrating the selection of study.

3.2. Risk of bias

The danger of researcher bias in the selection of the papers is reduced because the first search process was carried out programmatically. The sheer volume of studies found by the broad search queries shown in Fig. 1 and Table 2 provides enough research material for this SLR, although the possibility that pertinent manuscripts may not have been found by the automated approach is recognized.

To further mitigate potential subjectivity, the quality assessment was performed to evaluate the methodological rigor and relevance of the 99 included studies. Following the formal Quality Assessment (QA) procedures proposed by Kitchenham [14], the seven quality metrics detailed in Table 4 were applied. This systematic scoring ensures that the data synthesis is based on high-quality evidence, a core requirement of the Evidence-Based Software Engineering framework [13].

Table 5
The manuscripts filtered at the various stages broken down by editorial.

Stage	Process	Selection criteria	IEEE	ACM	Elsevier	Springer	MDPI	Others	Total
1	Identification	Initial search	802	165	507	358	170	928	**2930**
2	Identification	Filtering (Pre-screen)	771	161	501	329	169	760	**2691**
3	Screening	Title review	339	99	211	150	70	285	**1154**
4	Screening	Abstract review	128	69	55	46	22	52	**372**
5	Eligibility	Full-text assessment	42	20	15	8	9	5	**99**

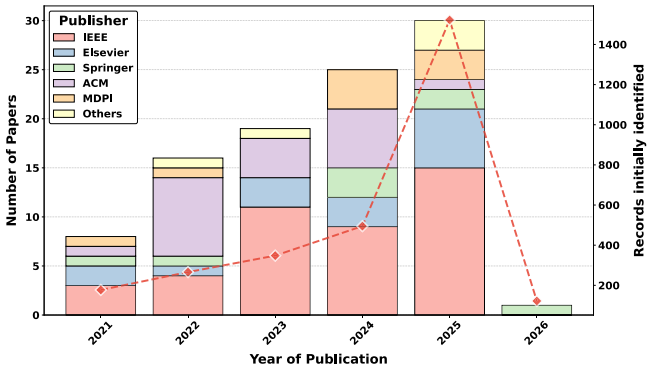

Fig. 3. Longitudinal analysis of the scientific production trends over the observed time period.

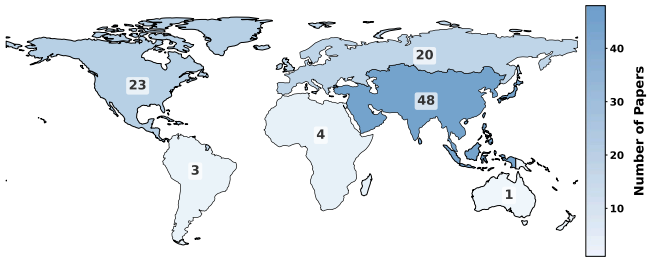

Fig. 4. Geographical distribution of scientific production based on author affiliations.

The analysis demonstrates a consistently low risk of bias across the selected primary studies, reflecting a high level of methodological soundness. A significant majority of the works exhibited strong validity, particularly excelling in scope alignment (**M1**) and the implementation of robust validation scenarios (**M2**). In addition, regarding the source selection strategy, non-academic records (gray literature) were intentionally excluded from this review. As noted in [23], the inclusion of gray literature is primarily recommended when the volume or quality of academic evidence is limited. Given that this research is supported by a substantial body of high-quality evidence, non-academic sources were excluded to ensure the review relies solely on rigorously peer-reviewed sources, consistent with the pursuit of high-integrity primary studies advocated by Kitchenham.

3.3. Replicability and data availability

To ensure the auditability and replicability of this SLR, as recommended by Kitchenham's guidelines [13,14], several measures were implemented. First, the search process was automated using a specialized client [17], which eliminates the variability of manual database interaction and ensures that the search string application is consistent and repeatable.

The replication package [24] explicitly documents the systematic filtering pipeline, providing a step-by-step trace of how the initial pool of 2930 records was refined to the final 99 primary studies. In accordance with Kitchenham's emphasis on transparency, the dataset includes the intermediate results of each selection stage: (i) the raw metadata from the initial automated search of 2930 records, (ii) the 372 records that remained after title and abstract screening, and (iii) the specific rationale for exclusion during the full-text eligibility assessment that led to the final 99 studies. Furthermore, the package provides the detailed quality assessment scores for each included study, ensuring the rigor and reproducibility of the quality metrics used in the evaluation.

3.4. Overview

The systematic selection process resulted in a final set of 99 primary studies. This section presents a bibliometric analysis of these selected articles to characterize the current state of the research field.

As illustrated in Fig. 3, the research interest in this domain has shown a consistent upward trend over the analyzed period. The number of publications rose steadily from 2021, peaking in 2025 with over 30 contributions, which indicates that the topic is currently experiencing its highest level of activity. A smaller number of papers are already recorded for 2026, reflecting the initial outputs of the current year. Regarding the publication venues, IEEE stands out as the dominant publisher, accounting for the largest share of papers across all years, particularly in 203 and 2025. It is followed by Springer and Elsevier, while contributions from ACM and MDPI maintain a stable but smaller presence.

The distribution of publication types reveals a highly balanced landscape between archival journal articles and conference proceedings. Journal articles represent the larger share of the selected studies ($n = 56$), slightly surpassing conference papers ($n = 43$). This distribution is particularly pertinent to this research, as publication type is utilized as a quality metric (defined in Table 4), where journal publications are favored due to their tendency to provide more extensive and insightful results. This context suggests that the field has reached a maturity level where rapid dissemination in conferences coexists with rigorously peer-reviewed journal contributions.

To analyze the global research activity, the primary studies were mapped based on the first author's affiliation. Fig. 4 highlights a significant concentration of research in Asia, which leads with 48 studies, reflecting a strong push for this technology in the region. North America and Europe follow with 23 and 20 studies respectively, showing comparable levels of activity, while contributions from South America ($n = 3$), Africa ($n = 4$), and Oceania ($n = 1$) are currently emerging.

Finally, an analysis of the most frequent keywords identifies the technological drivers of the field in Fig. 5. The results reveal a landscape heavily dominated by "Machine Learning", which stands out as the most frequent topic. This is followed by "Automated Program Repair" and "Multi-Agent Systems", which also constitute a significant portion of the field. "Edge Computing" further illustrate the focus on self healing in distributed systems. This distribution is pertinent to this research since topic relevance is a key selection criterion, favoring studies employing these advanced data-driven methodologies as they tend to offer more robust and scalable solutions than traditional approaches.

4. Discussion

In this section, the main findings derived from the systematic analysis of the 99 selected studies are discussed, with the aim of contributing

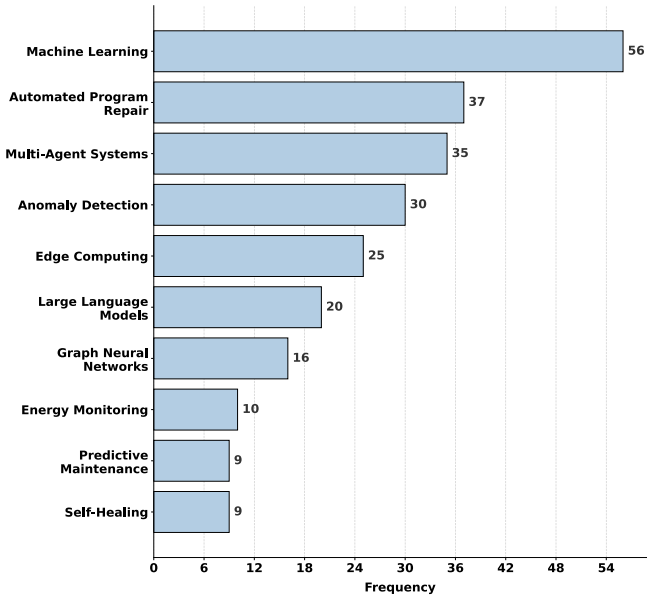

Fig. 5. Ranking of the top 10 most prominent keywords in the selected literature.

to the improvement of software development and maintenance. The objective is to synthesize the quantitative and qualitative evidence to provide comprehensive answers to the Research Questions defined in the protocol. Consequently, the following analysis elucidates the current landscape of the Autonomous Self-Healing in Distributed Continuum, mapping these findings to core software engineering areas: the predominant technical challenges in Section 4.1, the most widely adopted AI methods and techniques in Section 4.2, the underlying frameworks and architectures in Section 4.3, the associated ethical risks in Section 4.4, and the emerging future trends in Section 4.5.

4.1. RQ1: What are the main challenges in integrating AI-driven self-healing into distributed environments?

This research question seeks to examine the main challenges involved in combining two advanced concepts: AI-driven self-healing techniques and distributed computing environments. In settings such as multi-cloud, edge–cloud, or hybrid infrastructures, this integration becomes even more complex due to the heterogeneity of systems, the operational constraints of distributed locations, and the need for precise coordination across multiple layers. Understanding these challenges is essential for designing solutions that are reliable, scalable, and trustworthy. Fig. 6 summarizes the challenges encountered in the study.

4.1.1. Data management challenges

In the context of self-healing systems, the significance of data management cannot be understated. The type and quality of the data being gathered and processed represent one of the main obstacles to effectively combining intelligent monitoring with self-healing techniques [25,26]. Reliability problems [25,27], accessibility gaps [28], and inconsistencies in structure or semantics [29,30] are common problems with data. Accurate diagnostic or repair model training is quite challenging in many real-world scenarios due to the absence of labeled instances, particularly for uncommon failure conditions [31–33]. Additionally, learning systems may be biased toward false security or under-detection of crucial events due to the existence of noise [34], outliers [35], or imbalanced class distributions [36], where regular operations greatly outnumber abnormal events [10]. These biases can be further amplified by statistical phenomena such as Simpson's paradox [32] or the presence of spurious correlations [37], which may lead models to infer misleading relationships. Monitoring systems frequently collect enormous volumes of heterogeneous [38], multivariate data streams in real-time from a variety of sources, such as network traces, sensor data, application logs, and system metrics [29,39]. Downstream analysis can be made considerably more difficult by the complexity of such data, which arises from its various formats, large dimensionality, and complex interdependencies [40,41]. High-volume data environments, where high throughput does not always ensure sufficient data observability or actionable insight [42], only make this problem worse. Paradoxically, even in these large-scale datasets, data scarcity may arise for rare but critical events, creating substantial blind spots in model learning and validation [43,44]. Data localization is also a major challenge [37,45] since it takes a certain level of contextual awareness that is rarely simple to identify the precise root cause in a sea of noisy telemetry. Furthermore, privacy considerations may restrict the scope or granularity of data that can be monitored, resulting in gaps in repair logic and blind spots in detection [30,31,46,47]. In industrial environments, current systems suffer from rigidity and centralized architectures that create data processing bottlenecks, compromising real-time responsiveness and reliability [48]. Similarly, regarding data quality monitoring, traditional rule-based systems are often ill-suited for modern cloud data as they fail to detect complex semantic errors or schema drifts in unorganized data lakes [49].

RQ1 Finding #1: Effective self-healing is fundamentally constrained by data-related limitations, including inadequate data quality, scarce labeled instances for rare failures, and the complexity of large-scale, heterogeneous telemetry.

4.1.2. AI model challenges

Developing and deploying AI models that can support intelligent monitoring and repair adds another layer of complexity because it requires handling diverse data, ensuring accuracy, and deploying reliably [39,50]. Models must be generalizable enough to prevent overfitting to particular contexts or known failure patterns [51–53], but also flexible enough to accommodate different system types and data distributions [54]. Explainability is still a major issue, particularly in high-stakes or safety-critical systems, where decision-makers need to know why an automated diagnostic or repair recommendation is made before putting their trust in it [51,52,55]. Furthermore, for models to be accurate and feasible in real-world scenarios [37,56], which frequently entail drift in system behavior over time [26,46], they must contain continuous learning or periodic retraining [57,58]. This creates a trade-off between flexibility and training efficiency, especially in settings with limited resources [32,59,60]. At the same time, achieving models that are lightweight enough to operate within constrained edge or embedded systems, without sacrificing precision or robustness, remains a critical requirement [57]. However, the growing deployment of AI agents that autonomously detect and repair issues [50] introduces its own set of challenges, including ethical concerns [58], risks of unintended bias [40], fairness considerations [56], and the evolving role of humans in decision-making, known as human-in-the-loop (HITL) [61,62]. Evaluating these models is non-trivial, as traditional metrics may not fully capture the impact of a repair or the risk of an undetected failure [63].

The design of the reward function is crucial for RL systems should it become misaligned with actual operational objectives, it may result in undesirable behaviors [59,64]. To guarantee that solutions genuinely optimize the intended goals, evolutionary algorithms need carefully constructed fitness functions [65]. Prompt engineering is an essential ability for LLMs because of token restrictions that limit context length [54]. Also, traditional DL methods often rely on fixed prompts, which prevent the model from actively exploring the codebase to gather necessary 'repair ingredients' [66]. Numerous models exhibit significant hyperparameter sensitivity, necessitating a great deal of

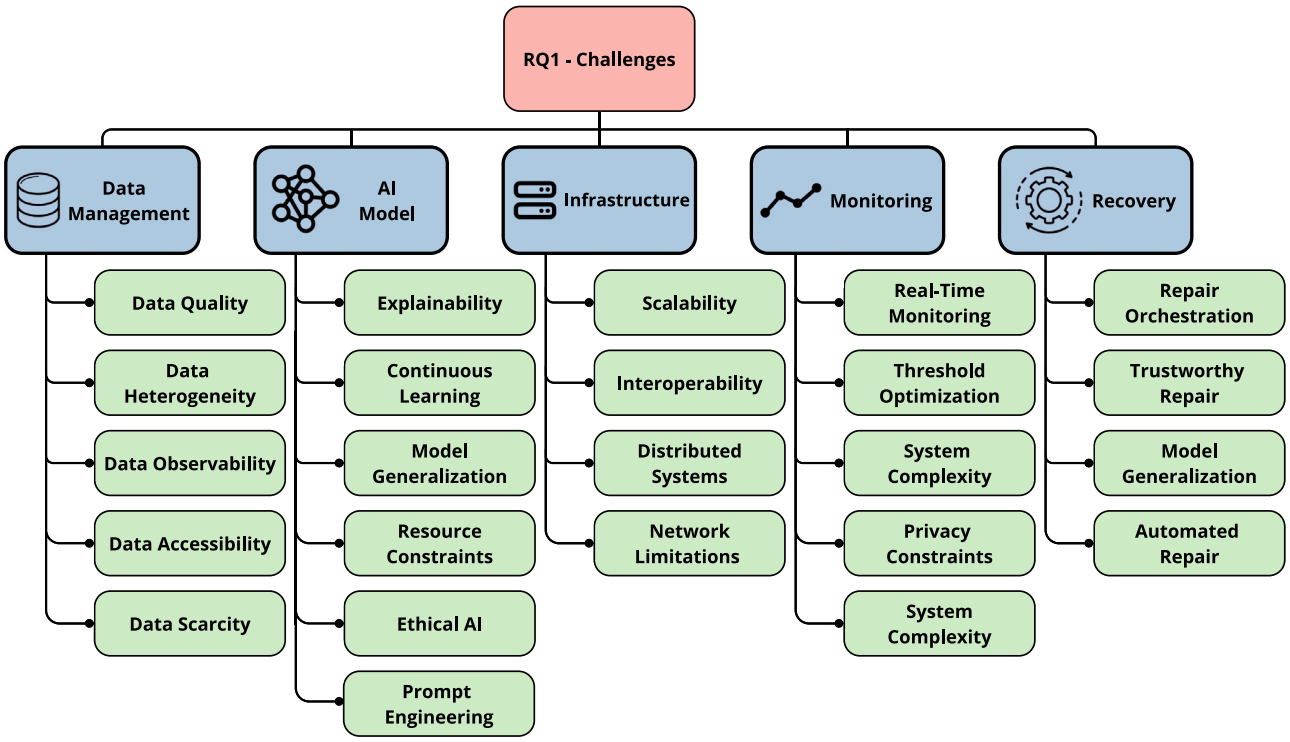

Fig. 6. An overview of the challenges associated with autonomous self-healing classified by category.

fine-tuning to get consistent and ideal performance [67,68]. Continuous learning systems run the risk of catastrophic forgetting, in which previously learned information is lost as a result of adapting to new patterns [47]. Context awareness is also essential for effective operation, especially for systems that must adjust to a variety of operating contexts. Besides, AI hallucinations can cause generative and retrieval-augmented systems to produce believable but inaccurate results [53]. Lastly, coordination, consistency, and evaluation get more difficult with combination or hybrid models that integrate neural techniques, statistical learning, and symbolic reasoning [27].

RQ1 Finding #2: The deployment of AI-driven self-healing models is constrained by challenges in generalization, explainability, continuous adaptation under drift, resource efficiency, ethical and evaluation concerns, prompting and the increasing complexity of hybrid and autonomous learning systems.

4.1.3. Infrastructure challenges

The deployment context for intelligent monitoring and self-healing systems is rarely ideal, often constrained by real-world infrastructure limitations that impact performance and scalability [69–71]. Large-scale systems, whether in cloud, edge, or hybrid environments, demand solutions that can scale efficiently across thousands of nodes while maintaining responsiveness and reliability [47,72]. At the edge, computational power may be limited, making it difficult to run complex models or repair engines without draining energy [59,64,73] or consuming critical system resources [8,10]. In cloud environments, however, the focus is often on designing systems that are cost-efficient while still delivering high accuracy, which remains a non-trivial engineering challenge [34,74]. Moreover, network limitations can hinder the transmission of data for centralized analysis, forcing trade-offs between local processing and remote orchestration [10]. Interoperability further complicates integration efforts, as intelligent monitoring and repair components often need to interface with legacy systems, diverse APIs, container platforms, and heterogeneous hardware stacks [25,50]. The lack of standardized interfaces or semantic alignment across these systems adds friction and increases the risk of failure during runtime coordination [46]. In addition, the presence of heterogeneous infrastructure with various architectures, operating systems, and deployment environments further complicates the implementation of consistent monitoring and remediation strategies [50]. Finally, fault isolation in distributed environments can be particularly complex, as cascading failures often obscure the true root cause [27,31,38], and coordinating self-healing actions across multiple infrastructure and application layers without introducing conflicts demands precise orchestration [47,70].

RQ1 Finding #3: Self-healing systems are constrained by infrastructure limitations related to scalability, resource availability, network constraints, interoperability, heterogeneity, and the complexity of fault isolation and coordinated remediation in distributed environments.

4.1.4. Monitoring challenges

Detecting or preventing system failures requires more than just collecting data; it also demands interpreting that data in real-time [51,64,75]. Effective monitoring systems need to identify anomalies not only accurately but also quickly and confidently, especially when supporting intelligent repair mechanisms [44]. While real-time detection is crucial for enabling prompt corrective action [67,76], many anomalies, particularly subtle shifts or emerging patterns, may not be visible during training or testing, which limits the model's effectiveness. Monitoring itself also introduces overhead, and as system complexity grows, analyzing monitoring data becomes increasingly challenging [76,77]. Techniques that rely on selective monitoring risk missing critical signs of failure as they try to reduce noise and overhead by focusing only on certain signals [33]. Setting an appropriate detection threshold remains a persistent challenge, as thresholds that are too strict can trigger excessive false alarms, while those that are too loose may allow critical anomalies to go unnoticed [33,35,72]. The task becomes even heavier when monitoring needs to cover user activity, system performance, security, and application behavior all at once [77,78]. Detecting and

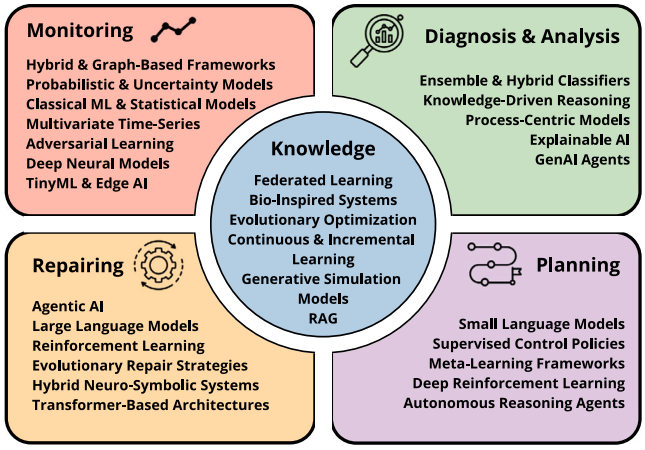

Fig. 7. Taxonomy of AI techniques integrated into the MAPE-K loop.

assessing group anomalies, where multiple components fail in a coordinated or cascading manner, is particularly difficult [29]. On top of that, privacy restrictions can limit how often or how extensively monitoring can be performed, especially in sensitive domains such as consumer Internet of Things (IoT), healthcare, or finance [35,46,53,73].

RQ1 Finding #4: Effective monitoring for self-healing is constrained by real-time interpretation challenges, anomaly detection trade-offs, overhead from complex systems, group and subtle failures, and privacy-driven limitations on observability.

4.1.5. Recovery challenges

Once an anomaly or fault is detected, the challenge shifts toward executing an effective, trustworthy repair [56,60]. Self-healing techniques must navigate a delicate balance between speed, safety, and completeness [47,56]. Setting detection thresholds that trigger a repair action without causing false positives is notoriously difficult [72], and the orchestration of multiple repair steps, especially across distributed systems or containerized environments, can become overwhelmingly complex [46,65]. Offloading repair logic to the cloud or centralized controllers introduces latency and dependencies [59], while keeping it local requires efficiency and minimal overhead [70]. The scope of what can be automatically repaired is also limited; many real-world failures involve business logic or context-specific errors that current repairing techniques cannot handle [68]. Trust remains a major barrier: operators may be reluctant to let a system apply a fix autonomously, particularly in production environments [40]. Finally, the repair process itself must be evaluated, establishing whether a repair truly restored correct behavior or merely masked a deeper issue [47]. These layers of complexity demonstrate that while self-healing systems hold great promise, their full realization demands overcoming significant technical, operational, and human-centric challenges.

RQ1 Finding #5: Recovery in self-healing systems is constrained by trade-offs between speed, safety, and completeness, the complexity of orchestrating multi-step or distributed repairs, limitations of automated fixes, latency and resource considerations, trust and HITL, and the challenge of verifying true restoration of correct behavior.

4.2. RQ2: What are the most effective AI-based approaches currently used for real-time monitoring of software systems and self-healing?

The analysis of the selected primary studies reveals a diverse algorithmic landscape where the effectiveness of an AI approach is intrinsically linked to the specific phase of the self-healing lifecycle it supports, as illustrated in Fig. 7. The findings indicate a clear technological progression: while resource constrained monitoring tasks at the edge still rely on lightweight classical Machine Learning and statistical methods, the diagnosis and execution phases are increasingly dominated by DL architectures and, more recently, LLMs and Generative AI. This transition reflects a trade-off between computational efficiency and semantic reasoning capabilities. The following sections categorize these approaches based on their functional role within the autonomic loop, detailing the most effective models for monitoring, diagnosis, decision-making, adaptation, knowledge integration, and code repair.

4.2.1. Monitoring and fault detection models

The monitoring and fault detection phase of self-healing systems have seen the application of a broad range of artificial intelligence approaches, integrating sophisticated DL, hybrid architectures, and classical machine learning to detect anomalies and anticipate possible breakdowns. Conventional methods are still useful for lightweight defect identification, especially in circumstances with limited computational resources like fog and edges. For anomaly classification and predictive maintenance, supervised classifiers such as Support Vector Machines (SVM) and their variations, like One-Class SVM, are used because of their low processing cost and robustness [33,45]. Besides, the simulation of sequential dependencies in edge clusters, probabilistic models like Hidden Markov Models aid in the detection of temporal anomalies [45]. Moreover, to support fault prediction and service disruption classification, decision trees and random forests improve interpretability, being used to detect cybersecurity or communication problems [26,45,79]. For fault detection in unsupervised clustering tasks, methods such as Isolation Forests, K-Means, and DBSCAN are employed as they can isolate outliers or identifying anomalous points that are not a part of any dense area [26].

Deep neural architectures significantly enhance detection accuracy by capturing non-linear dependencies and spatiotemporal correlations. Autoencoders and LSTM-Autoencoders perform unsupervised reconstruction-based detection, learning normal operational patterns and flagging deviations as anomalies [38,80]. Variants such as the Spatiotemporal Autoencoder with ConvLSTM2D and Center-Weighted Loss focus on video-based anomaly detection, emphasizing central regions to minimize false positives [73]. Convolutional neural networks and similar architectures are used to forecast temporal faults in time-series data because they can improve the accuracy of anomaly detection in dynamic situations by capturing short-term patterns and local relationships in sequential signals [36,64]. Other architectures such as LSTM networks [10] and GNN-GRU hybrids [27] capture both temporal and spatial dependencies, enabling predictive failure detection and root-cause localization in distributed systems. In the industrial IoT domain, Two-Phase Dual-Adversarial Agents model [29] and decentralized supervised learning model [30] employ cooperative and adversarial autoencoder training to achieve fast, unsupervised detection on resource-limited devices. Additionally, in [81], TinyML neural network performs real-time fault diagnosis on edge devices by classifying motor conditions. A kernel-based non-parametric regression method has also been applied in [39] to predict tool wear efficiently on edge devices, leveraging similarity-driven attention to highlight relevant sensor inputs for prediction. In photovoltaic systems, ANN achieves the best overall performance for anomaly detection, with KNN and LOF performing best in unsupervised scenarios [63,75].

Recent research has introduced advanced composite frameworks like CGNN-MHSA-AR, which integrate Graph Neural Networks, multi-head self-attention, GRUs, and autoregressive components to enable explainable anomaly detection in cloud environments [82]. Alongside these, hybrid and attention-driven models have also emerged, such as MTAD-GAT, an unsupervised graph-based architecture that combines CNNs and LSTMs for detecting anomalies in multivariate time-series data [51], and LogEDL, a semi-supervised Transformer-based model

that leverages Evidential DL to quantify predictive uncertainty when identifying previously unseen log anomalies [52]. Finally, explainable extensions such as fastSHAP-C support real-time interpretability of deep anomaly detectors, enabling HITL understanding of detected failures [83].

RQ2 Finding #1: Monitoring and fault detection in self-healing systems utilize a range of modeling paradigms, including classical machine learning, probabilistic models, deep neural architectures, hybrid graph and attention-based frameworks, and resource-efficient TinyML models for real-time edge deployment.

4.2.2. Diagnosis and root-cause analysis models

In order to enable well-informed recovery decisions, diagnosis and root-cause analysis models seek to identify the underlying causes of anomalies that have been found and offer interpretability. Explainability techniques such as SHAP and TreeSHAP are widely applied to quantify the contribution of each feature to a model's output, enabling a better understanding of which variables most influence fault occurrence [79]. Additionally, fastSHAP-C presents a reactive explainable AI method that combines interpretability and low computing cost to provide real-time explanations of deep autoencoder decisions in anomaly detection tasks [83]. Other works employ process-centric methods, including Process Mining and Token Replay, to detect behavioral anomalies and identify deviations from expected workflows through conformance checking and rule-based evaluation [84]. Hybrid and ensemble frameworks further enhance diagnostic performance; for instance, the Dynamic Selection Multiple Classifier System combines anomaly detection, model selection, and ensemble prediction to achieve adaptive resource usage prediction and fault diagnosis [28]. Similarly, in [85], interpretable tree-based models such as Ordinal Decision Trees are used for root-cause analysis in industrial environments. This model provides transparent decision logic that helps human operators understand failure patterns. Knowledge-driven approaches like Case-Based Reasoning complement these data-driven method by using historical fault cases to infer likely causes of current issues. In [86], the authors apply a similar reasoning to the software domain by combining ALBERT, DRMM, and collaborative filtering to localize bugs based on the semantic and contextual similarity between bug reports and source code. Moving toward fully autonomous operations, next-generation Agentic AIOps frameworks leverage Generative AI and Agentic AI to drive hyper-automation, evolving beyond simple monitoring to enable autonomous diagnosis and incident resolution without human intervention [87].

RQ2 Finding #2: Diagnosis and root-cause analysis in self-healing systems employ explainable AI techniques, process-centric methods, hybrid and ensemble frameworks, GenAI Agents and knowledge-driven approaches to identify underlying causes and support interpretable recovery decisions.

4.2.3. Recovery and decision-making models

Recovery and decision-making models form the operational core of self-healing systems. Their main goal is to determine and execute the most suitable corrective actions once faults have been detected and diagnosed. RL techniques play a central role in this stage because they allow systems to learn recovery policies by interacting with their environment and evaluating the impact of each action. Models such as Proximal Policy Optimization [64] and Q-Learning [88] have been widely adopted to manage resource allocation and reconfiguration processes, restoring system stability while maintaining performance. ReLIEF [45] extends this idea to fog environments, where it dynamically assigns tasks according to resource availability and network conditions to ensure reliable operation. More advanced approaches, including Deep Recurrent Q-Networks [59] and RSD4 [42], integrate recurrent neural architectures to deal with temporal dependencies and partial observability in complex, dynamic systems. These models optimize decisions such as task offloading, CPU scheduling, and energy management under changing conditions. In a similar direction, RDDNR-DQN and RDDNR-SAC [65,88] apply Deep Reinforcement Learning (DRL) to power grid management, using reward-based optimization to reconfigure networks efficiently and improve recovery times. In cybersecurity, DRL is utilized to optimize automated incident response and threat mitigation strategies through trial and error in simulated environments [89].

Recent studies emphasize decentralized and agent-based recovery to handle complexity and scale. In vehicular edge computing, Multi-Agent Reinforcement Learning frameworks like MAPPO-GAT address competitive resource allocation and partial observability [90], while Federated MARL with Potential Fields optimizes task offloading while preserving privacy [91]. For power grid restoration, Decentralized Parallel Mesh-based Minimum Spanning Tree (DPMST) algorithms allow zone agents to execute concurrent recovery processes [92], and Distributed Agent Controllers (DACs) employ sensitivity-based gradient methods to maintain voltage stability [93]. Addressing resource constraints, in [94], Small Language Models (SLMs) are used to execute complex recovery tasks securely, offering an efficient alternative to heavy LLMs.

Beyond RL, many works incorporate supervised and meta-learning methods to enhance adaptability and reduce the need for retraining. Deep neural networks can directly learn the relationship between the system's current state and the optimal recovery action, enabling fast and data-driven decision-making [50]. In turn, actor–critic architectures refine these recovery actions through continuous feedback from previous interventions, improving the reliability of future decisions. Meta-learning approaches such as MAML-KAD [57] and Meta-APR [77] take this adaptability further by allowing models to generalize across different failure types and to learn new recovery strategies with very limited data. This capability is especially valuable in dynamic environments, where faults evolve quickly and labeled training data are scarce. Moreover, large-scale reasoning models like the ReAct agent built on GPT-4 [61] illustrate a new direction in decision-making, where language models autonomously reason, plan, and execute recovery actions in cloud systems.

RQ2 Finding #3: Recovery and decision-making strategies in self-healing systems rely heavily on RL to optimize corrective actions. These are increasingly complemented by decentralized agent-based frameworks for scalability, meta-learning for rapid adaptability to few-shot failure scenarios, and LLMs and SLMs that enable autonomous reasoning and planning in complex environments.

4.2.4. Adaptation and optimization models

Self-healing systems can dynamically modify their behavior in response to changing conditions thanks to adaptation and optimization models. To preserve stability, performance, and resilience, these models optimize system parameters, retrain learning components, and continuously improve configurations. Numerous methods emphasize incremental and continuous learning, enabling models to update themselves in response to fresh data. Systems can identify changes in operational behavior and modify their predictive or decision-making models without requiring complete retraining by using strategies like incremental learning and drift detection [46]. When workloads, network conditions, or resource constraints change over time, this constant adaptation keeps performance from degrading.

Additionally, optimization techniques are essential for promoting self-healing behavior. In [46], the configuration space is explored and the most effective infrastructure settings for service deployment and maintenance are identified using a metaheuristic approach that uses NSGA-II, SPEA2, MOEA/D, SMPSO, and MOCell. Similarly, a heuristic optimizer called RPFHO is used in [80] to automatically adjust

the DL models' hyperparameters, increasing accuracy and stability while lowering the amount of manual intervention. Furthermore, in data quality monitoring, Q-learning agents autonomously learn and select the best remediation strategies over time, adapting to new error patterns without manual rule updates [49].

Adaptive frameworks support large-scale resilience by integrating distributed intelligence and self-organizing principles in addition to optimization. This strategy is demonstrated by the Self-aware Distributed DL model in [95], which maintains learning performance across diverse IoT environments, balances workloads, and dynamically assigns roles.

In [96], Artificial Immune System simulates adaptation through immune mechanisms like clonal selection and immune memory, is a particularly biologically inspired approach. Over time, AIS-based systems identify abnormalities as "pathogens", learn typical behavior patterns, and adjust recovery strategies.

RQ2 Finding #4: Adaptation and optimization in self-healing systems leverage incremental learning, drift detection, metaheuristic optimizers, and autonomous Q-learning strategies, alongside distributed self-organizing frameworks and biologically inspired models like Artificial Immune Systems, to dynamically adjust system behavior, maintain performance, and enhance resilience.

4.2.5. Learning from experience and knowledge integration models

Models for integrating knowledge and learning from experience allow self-healing systems to capture, share, and reuse what they have learned from past events. Instead of simply reacting to faults, these models build a kind of collective intelligence that improves decision-making across devices, system layers, and over time.

A key part of this is federated learning. In this approach, distributed devices can train models locally, such as FedLSTM or FedSVM [32], and share only the model parameters, not the raw data. This decentralized setup preserves privacy, scales efficiently, and supports real-time tasks like anomaly detection and remaining useful life prediction in industrial IoT environments. Frameworks like BytoChain and CBFL [45] take this further by using proof-of-accuracy consensus and Byzantine fault tolerance, ensuring that the global learning process is resilient against faulty or malicious updates. Similarly, CRACAU [45] improves model reliability by detecting and excluding corrupted contributions during collaborative training.

Knowledge integration also plays a key role in supervised and generative models that predict system degradation and enhance fault prevention. For instance, CAROL [45] applies supervised learning to forecast Quality of Service (QoS) and dynamically fine-tune model parameters, while PreGAN [45] leverages generative DL to simulate the impact of task migration before it happens—helping prevent resource exhaustion and node failures. Complementing these data-driven methods, knowledge retrieval systems like Big Data Search [97] collect relevant past cases and patterns from historical repositories to support better diagnosis and adaptive decision-making.

Modern self-healing systems are increasingly aligned with the RAG, agent-based, and agentless paradigms [98]. By balancing autonomous planning with precise context retrieval, these paradigms allow models to move beyond reactive fault correction toward a collective intelligence that improves decision-making across all system layers.

RQ2 Finding #5: Learning from experience and knowledge integration in self-healing systems employs federated learning models with resilient aggregation frameworks, supervised and generative predictive models, RAGs and knowledge retrieval systems to capture, share, and reuse operational insights for improved diagnosis, adaptation, and fault prevention.

4.2.6. Repair and self-healing execution models

Repair and self-healing execution models represent the most advanced stage of the self-healing lifecycle, the point where a system can autonomously generate, validate, and apply fixes to restore its own functionality. Recent research in this area is largely driven by Transformer-based architectures. Models like CodeT5 and CodeT5-large can generate code fixes directly from buggy inputs using encoder–decoder trained on massive datasets of defective and repaired code [8,44]. Their deep contextual understanding allows them to produce precise, minimal patches tailored to specific failure scenarios. Complementary LLMs, including GPT-4, ChatGPT, Codex, GPT-Neo, CodeGPT and InCoder [44,56,62,99], extend these capabilities even further. By combining generative and infilling techniques, they can repair incomplete or erroneous code snippets with impressive accuracy and adaptability.

Hybrid frameworks push this concept forward by integrating multiple reasoning paradigms to improve the reliability of automated repairs. For example, SGEPR [60], and VRepair [100] blend Transformer architectures with graph neural networks, rule-based reasoning, or pointer mechanisms. This helps capture both the syntactic and semantic relationships within code, leading to higher-quality and more interpretable patches. Meanwhile, the VulAdvisor framework highlights the expanding relationship between LLMs and secure software maintenance by combining CodeT5 and ChatGPT through attention-based refinement and prompt engineering to create vulnerability-aware patches [40]. GrasP [101] leverages abstract syntax tree representations and a Graph2Seq neural network with attention to learn fix patterns from buggy-fixed method pairs and generate automated code repair patches.

A significant evolution in self-healing is the rise of Agentic AI, where autonomous systems go beyond static code generation to actively plan and execute repairs. In [66], Finite State Machine (FSM) is used to guide the LLM through a structured lifecycle of information gathering, tool execution, and validation. To address the complexity of large codebases, in [102] leverages Monte Carlo Tree Search to navigate and explore repository-level contexts efficiently. Furthermore, in [103] employ Environmentally RL to dynamically coordinate specialized agents, ensuring more precise and context-aware resolutions than single-model approaches.

In addition to these architecture-driven techniques, other strategies use the immune and biological systems as models to promote ongoing healing. The Artificial Immune System [96] views fault management as an immune response, learning typical behavior, and gradually developing corrective measures. Similarly, big generative models like GPT-4 and Code Llama [104] can generate validation tests and code fixes on their own, thereby completing the feedback loop between fault detection and verification.

RQ2 Finding #6: Repair and self-healing execution in advanced systems primarily relies on LLMs, hybrid frameworks combining transformers with graph neural networks, rule-based reasoning, or pointer mechanisms, and biologically inspired approaches such as Artificial Immune Systems to autonomously generate, validate, and apply corrective actions.

4.3. RQ3: What architectures, frameworks, and tools support self-healing while enabling the seamless integration of AIOps with MLOps systems?

This section focuses on the technical pillars that enable self-healing capabilities within integrated AIOps and MLOps environments. To achieve a system that can autonomously detect, diagnose, and remediate failures, three critical functional layers must work in tandem: AI-Driven Monitoring for real-time observability and anomaly detection, AI-Driven Repairing for automated fault correction and code

Table 6
Taxonomy of architectures and frameworks for self-healing in AIOps/MLOps systems.

Category	Focus area	Methodology/Technique	Ref.
AI-driven monitoring frameworks	Edge & IoT Ops	IoT Data + ANN; Lightweight Kernel Regression	[39,63,64]
		TinyML & Temporal CNN (On-device detection)	[36,81]
		Federated Learning (Privacy-preserving Edge)	[10]
	Cloud & software	Multi-modal: Logs, Metrics, Traces	[27]
		Masked LM + Evidential DL	[52]
		Isolation Forests (Data Lake Quality)	[49]
	Security	Adaptive Multi-Agent RL	[71]
		Agentic AI + DRL	[89]
AI-driven repairing frameworks	Code repair (LLMs)	LLMs with Logic Explanation & Reasoning	[40,105]
		Automation + Reasoning	[106]
		Transfer Learning (Vulnerability Fixing)	[100]
		Graph-based Context Matching	[68,101]
	Agentic repair	Monte Carlo Tree Search	[102]
		Finite State Machine Agents	[66]
		Co-Learning Multi-Agent Frameworks	[103]
	Physical systems	Deep RL Agents	[65]
		RL for Hardware Physical Design	[107]
	Model repair	Self-Organizing Distributed DL Frameworks	[95]
		Differential Evolution (Weight repair)	[108]
AI-driven scheduling frameworks	Orchestration	Agentic AI + Declarative Configurations	[109]
		Sensing & Reasoning agents	[48]
		Token-based Self-Organizing Orchestration	[95]
	Resource optimization	Multi-Objective Optimization (IaC)	[46]
		AI-Driven Scheduler + Reinforcement Learning	[50]
		Deep RL + RNN	[42]

rectification, and AI-Driven Scheduling and Orchestration for maintaining service continuity through dynamic resource management. The following sections detail the architectures and frameworks identified in the literature, categorized by their primary role in the self-healing lifecycle. Table 6 provides a structured taxonomy of the tools and frameworks identified in the literature, mapping specific methodologies to their respective functional roles.

4.3.1. AI-driven monitoring tools

Monitoring technologies are increasingly applied across multiple domains, leveraging advanced machine learning and edge computing techniques to enhance reliability. In the renewable energy IoT domain, the authors in [63,64] demonstrate that predictive algorithms combining IoT data with Artificial Neural Networks enable efficient energy management, reduce operational costs, support predictive maintenance, and provide real-time monitoring, ultimately improving the reliability of photovoltaic solar systems. Meanwhile, in the Industrial Internet of Things (IIoT), some authors are implementing Graph Attention Networks and MLFlow to build explainable, scalable, and reliable fault detection systems [51]. Other approaches use a dual-agent setup that allows models to learn both accurate reconstruction of normal data and amplification of reconstruction errors when anomalies occur [29]. Lightweight predictive models, such as Nadaraya–Watson kernel regression [39] and Tiny ML techniques [81], have been proposed to achieve high predictive accuracy while remaining fast, resource-efficient, and suitable for real-time edge condition monitoring. Newer approaches emphasize decentralized intelligence to handle data volume. For instance, in Smart City infrastructures, a 5G Edge AI framework utilizing Adaptive Multi-Agent Reinforcement Learning allows data to be processed locally, reducing latency by over 80% while preserving privacy [71]. Moreover, unsupervised distributed feature selection using MST-based clustering helps reduce network latency and bandwidth usage, improving fault detection accuracy while preserving data privacy by keeping raw data local [37].

Building on these concepts, edge computing applications benefit from lightweight Temporal Convolutional Neural Networks, which enable real-time anomaly detection directly on devices without centralizing data [36]. Similarly, as noted in [10], federated learning in 5G IoT networks enables training directly on edge devices using local data, thereby reducing communication overhead and preserving privacy. These methods minimize edge resource usage by reducing model size, making them ideal for resource-constrained environments.

Complementing edge-based solutions, cloud computing plays a critical role in large-scale monitoring and analytics. In [27], the authors propose that system failures in microservices can be predicted, and their root causes identified by integrating logs, metrics, traces, and events using a GNN-GRU model, which improves prediction accuracy and enables rapid root cause localization [27]. Additionally, real-time monitoring combined with continuous learning replaces static, rule-based systems with adaptive machine learning models capable of dynamically detecting and responding to cyber threats [26], while for secure and reliable system monitoring, other studies integrate DL with blockchain technology to ensure low-latency anomaly detection while preserving privacy and integrity [31]. Federated learning methods in distributed cloud environments further enhance anomaly detection accuracy, reducing false positives and preserving user privacy, while the system adapts to changing conditions without losing robustness [38]. In this context, LogEDL combines masked language modeling with evidential DL to detect log anomalies, quantifying prediction uncertainty with Dirichlet distributions for robust identification of both known and unknown anomalies [52].

Addressing data quality, recent studies introduce autonomous frameworks using Isolation Forests and Autoencoders to detect semantic errors and schema drifts in data lakes, actively triggering self-healing pipelines [49]. Security monitoring has also evolved; for example, a self-healing system integrated with Agentic AI and Deep Reinforcement Learning (DRL) can now autonomously isolate compromised nodes and update firewalls [89]. Furthermore, two-phase reliability models integrated with AI-driven modules for anomaly detection in [50], enables the cloud recovery without wasting any resource.

RQ3 Finding #1: AI-driven monitoring tools are evolving into decentralized, self-healing ecosystems by shifting intelligence to the edge with lightweight, privacy-preserving models, these systems minimize latency and resource usage while enabling effective anomaly detection and predictive maintenance. The incorporation of Agentic AI and DRL empowers these tools to transcend passive observation, allowing them to autonomously isolate security threats, identify root causes, and execute self-healing protocols without human intervention.

4.3.2. AI-driven repairing tools

Patch quality, developer trust, and efficiency have all consistently improved with recent developments in code rectification using LLMs. Systems like those in [40,62,105], for example, demonstrate how LLMs may produce patches while providing an explanation of their logic, enhancing confidence, decreasing manual labor, and assisting developers with code review and bug fixes. Similarly, by combining automation and reasoning, InferFix [106] improve patch quality and reliability, surpass baseline methods, decrease manual intervention, and increase the overall effectiveness of bug detection and repair. Patch correctness assessment pipelines [54] increase the dependability of APR tools across projects, while complementary efforts like Denchmark [97] offer high-quality datasets connecting bug reports to fine-grained buggy entities, enabling reproducible studies and lowering costs in debugging research. Similarly, SBugLocater [86] uses semantic and relevance matching layers to improve bug localization accuracy, and VRepair [100] uses transfer learning to improve token-level vulnerability fixes. Furthermore, richer syntactic and contextual information is captured by graph-based methods like GrasP [101] and Katana [68], enabling more accurate and credible patch generation across a variety of issue kinds.

The field is currently shifting towards Agentic systems that demonstrate repository-level reasoning. The LingmaAgent framework [102] uses Monte Carlo Tree Search to explore complex repositories, achieving significant improvements over earlier tools like SWE-agent. Similarly, RepairAgent [66] employs a Finite State Machine to dynamically interleave testing and validation tasks. This autonomy is further validated by Passerine [110] on enterprise-grade bugs, and by concepts like Agentic Translation [111] for rectifying logic errors. In educational contexts, multi-agent frameworks like Co-Learning [103] now dynamically select specialized agents to correct code errors. Expanding into hardware automation, OpenROAD Agent [107] introduces a self-correcting framework that utilizes RL to autonomously generate and refine physical design scripts, effectively mitigating LLM hallucinations in domain-specific tasks.

In the field of renewable energy IoT, the RDDNR framework presented in [65] employs DRL agents to autonomously and efficiently restore service after faults by selecting optimal switch configurations, thereby accelerating recovery while maintaining grid stability and efficiency. In [88], the integration of RL with context-oriented programming enables systems to automatically learn and apply corrective actions when similar issues recur. This approach allows systems to adapt dynamically, recover from unexpected failures, and continuously improve over time. Meanwhile, in the IIoT domain, system architectures typically comprise modules for control, diagnostics, forecasting, and external communication. By integrating machine learning and case-based reasoning, in [85], the system can perform predictive maintenance, detect anomalies, and even achieve self-healing.

Recent developments in Deep Neural Network (DNN) model correction include an emphasis on targeted repair techniques as well as architectural flexibility. Without depending on centralized cloud resources, a self-aware distributed DL framework designed for heterogeneous IoT edge devices integrates dynamic self-organizing and self-healing methods to improve system scalability and reliability. This approach allows for real-time reconfiguration and resilience in training tasks across diverse hardware platforms [95]. In addition, the authors of [108] provide Arachne, a search-based repair method that uses differential evolution to directly modify neural weights to fix particular misclassifications without requiring retraining. For targeted DNN corrections, including fairness changes in biased models, it provides a workable option.

RQ3 Finding #2: AI-driven repairing tools are evolving into autonomous, agentic systems capable of deep reasoning across software, physical infrastructure, and neural models. These tools move beyond simple patching to autonomously navigate complex repositories, self-heal energy grids, and directly repair neural network weights without the need for human intervention or costly retraining.

4.3.3. AI-driven scheduling and orchestration tools

In modern cloud and edge environments, intelligent scheduling and orchestration are essential for enabling self-healing capabilities. AI-based tools ensure systems can autonomously detect failures, adapt to changing conditions, and maintain service reliability. Recent literature defines a clear trend toward Agentic AIOps. Novel frameworks now extend the classic MAPE-K cycle with Agentic AI to execute declarative configurations autonomously [109]. Similarly, the Autonomous Intelligence Maturity Framework [48] guides the transition to self-organizing ecosystems using specialized Sensing and Reasoning agents, aligning with ITIL standards for heterogeneous environments [87].

In [50], the authors propose a service-oriented reliability framework that divides cloud operations into request processing and execution phases. AI-driven schedulers dynamically adjust resources based on demand, while anomaly detection and RL refine recovery strategies over time. Furthermore, the PIACERE framework, introduced by the authors in [46], leverages multi-objective optimization techniques for Infrastructure-as-Code (IaC), enabling automated deployment and reconfiguration across heterogeneous environments. It incorporates self-learning and anomaly detection mechanisms to trigger corrective actions, ensuring that the infrastructure adapts to failures without requiring manual intervention. In [42] present RSD4, a DRL algorithm for delay-constrained scheduling. It uses recurrent neural networks to handle partial observability and learns optimal policies under dynamic conditions. Techniques like user-level decomposition and node-level merging ensure scalability in large and multi-hop systems. Additionally, in [95], authors introduce a framework for orchestrating DL tasks across heterogeneous IoT edge devices, Self-Aware Distributed DL. It features dynamic task assignment, token-based self-organizing orchestration, and self-healing scheduling to maintain training continuity without human intervention.

RQ3 Finding #3: AI-driven scheduling and orchestration are converging into Agentic AIOps, where systems utilize Sensing and Reasoning agents to autonomously manage the lifecycle of cloud and edge resources. By embedding DRL, these tools enable self-organizing ecosystems capable of dynamic task assignment and self-healing. This allows for real-time adaptation to failures and partial observability in heterogeneous environments, ensuring operational continuity without manual intervention.

4.4. RQ4: What are the potential risks and ethical implications of using AI-driven techniques for system monitoring and program repair in critical applications?

As AI-driven techniques become deeply integrated into system monitoring and APR, their role transitions from simple optimization tools to critical decision-makers. In high-stakes applications—such as industrial control systems, autonomous infrastructure, and large the deployment

Table 7
Taxonomy of ethical risks in the associated literature.

Category	Identified problem	Ref.	Mitigation
Privacy	Data leakage Identity exposure Sensitive attributes	[10,67,105] [73] [27,58]	Federated Learning, differential privacy, homomorphic encryption, and GDPR compliance [10,30–32,58,71,73,91,105,112].
Transparency	Black-box model LLM hallucinations Lack of interpretability	[83,89,113] [53,105] [53,105]	XAI (SHAP, fastSHAP), reasoning traces, and HITL validation [27,79,83,105,109,113].
Bias & Equity	Data imbalances Domain shift & exposure bias Discriminatory decision-making	[53] [43] [58,114]	Adversarial debiasing, fair representation learning, curriculum learning and credit allocation mechanisms [43,53,58,91].
Security	Poisoning & evasion Silent patch errors Overfitting in repairing Expanded attack surface	[53] [44,67,105] [108] [49,112]	Secure training protocols, Post-Quantum Cryptography, and self-healing resilience [67,94,108,115].
Generalization	Data heterogeneity Rigid template	[10,27] [44,67]	Transfer learning, domain adaptation, and segmented prompting [10,43,55,67].
Governance	No human intervention Divergence from business logic	[44,114] [105]	Expert validation, manual review, autonomic thresholds and feedback loops [76,105,114].
Regulation	Lack of standardized frameworks AI weaponization	[25] [58]	Standardized monitoring, regulation-driven design, and policy alignment [25,58,73].
Operational	Inaccurate maintenance Precision vs. Privacy	[116] [32]	Non-intrusive sensing and weight optimization strategies [32,90,108,116].

of these technologies introduces a complex landscape of ethical and operational risks.

While AI offers unprecedented speed in detecting anomalies and patching vulnerabilities, it also brings systemic challenges: from the "black-box" nature of DL models to the risk of hallucinations in LLMs. Addressing RQ4 requires an exploration of how privacy, transparency, and accountability can be maintained when human oversight is potentially bypassed by automation. The following sections detail the primary ethical dimensions identified in current research, categorized by the nature of the risk and the proposed technical or procedural mitigations. To provide a structured overview of these challenges, Table 7 provides a structured summary of these ethical dimensions, categorizing the core challenges identified in the literature along with their respective mitigation strategies.

4.4.1. Privacy protection and responsible data use

One of the most pervasive ethical challenges in AI-enabled monitoring systems is the preservation of data privacy. For instance, the reliance on single data modalities may inadvertently expose sensitive attributes [27], while continuous video surveillance poses severe risks of revealing personal identities [73]. Moreover, the exposure of raw data during federated model updates threatens confidentiality, and cloud-based infrastructures further amplify concerns as providers may gain access to sensitive device-level information [10]. Similar privacy risks emerge in distributed intelligence exchange frameworks [58], where breaches could compromise critical infrastructures, and in LLM-based systems, where training on benchmark datasets may result in unintended data leakage or biased evaluations [67,105]. To mitigate these risks, diverse strategies have been proposed: Federated Learning to avoid sharing raw data across nodes [30–32,58,73], differential privacy, secure aggregation, and homomorphic encryption to protect updates during training [10,58], and local data processing with GDPR compliance to limit external transmission of sensitive video streams [73]. Complementary approaches such as blockchain for data integrity [31] and benchmarking frameworks like ARHE to prevent training data contamination [105] further strengthen privacy protection. Recent studies show that Federated Learning (FL) is essential for reducing risks in smart networks. FL protects privacy by training models without sharing raw user data [71,91]. Furthermore, in healthcare and critical infrastructure, combining FL with Post-Quantum Cryptography has been proposed to secure communications against future quantum computing threats while preserving data privacy [112].

RQ4 Finding #1: To mitigate privacy risks in AI monitoring and LLM systems, researchers employ strategies like federated learning, differential privacy, and blockchain to ensure data confidentiality and compliance.

4.4.2. Transparency and explainability as foundations for trust

The black-box nature of AI models presents a fundamental obstacle to accountability, as it hinders users from understanding how predictions are generated [83,113]. Lack of explainability in DL and LLM-driven debugging further undermines trust, since hallucinations and opaque reasoning may yield misleading or insecure outputs [53,105]. To address this, several studies advocate the adoption of explainable AI (XAI) techniques, such as SHAP and fastSHAP, which provide both local and global feature attributions to clarify decision-making processes [27,79,83,113]. Other works propose integration of execution evidence, reasoning traces, and HITL validation to ground explanations in verifiable results [105]. Such approaches not only improve transparency but also enhance accountability in AI-assisted decision support. In machinery health monitoring, the lack of adaptability in black-box systems is a major hurdle; researchers argue that XAI is essential here to provide transparent, tailored explanations to operators, fostering effective human-machine collaboration [109]. Future research in self-healing security systems also prioritizes increasing transparency in decision-making to ensure robustness against adversarial attacks [89].

RQ4 Finding #2: To combat the distrust caused by opaque black-box models, researchers advocate for XAI techniques and HITL validation to ensure transparency.

4.4.3. Bias, fairness, and equity in self-healing AI

Bias in training data remains a central ethical concern, as imbalanced datasets can skew model performance and reinforce unfair outcomes [53]. Domain shifts and limited adaptation datasets exacerbate this issue by leading to exposure bias and under-representation of certain classes [43]. In addition, AI systems deployed in critical infrastructures face risks of discrimination and unfair decision-making if these biases are left unaddressed [58]. In multi-agent systems, fairness extends to the contribution of individual agents; frameworks employ credit allocation mechanisms to ensure contribution fairness while preserving agent individuality [91]. Additionally, algorithmic bias remains

a cited ethical concern in the integration of Agentic AI, necessitating rigorous checks to prevent discriminatory outcomes in autonomous operations [114]. To mitigate such challenges, researchers have proposed fair representation learning, adversarial debiasing, and curriculum learning to enhance generalization [43,58], as well as data replication and advanced loss functions to correct imbalances in training datasets [53]. These techniques collectively aim to strengthen model robustness while ensuring fairness and equity across diverse operational contexts.

RQ4 Finding #3: Employing methods such as curriculum learning, data replication, adversarial debiasing and advanced loss functions guarantees fair and reliable performance.

4.4.4. Security, robustness, and the risk of exploitation

Ensuring robustness and security is a critical ethical concern in self-healing AI systems. Models are vulnerable to adversarial attacks such as poisoning and evasion, which can mislead predictions and compromise operational integrity [53]. Risks extend to over-reliance on AI in critical infrastructure, where failures may have cascading consequences [105], and to automated patching systems, which may introduce silent errors or security flaws [44,67]. Furthermore, optimization-driven repair strategies risk overfitting and unintended side effects if not carefully controlled [108]. The transition of critical infrastructures to digital formats significantly expands the attack surface, making them vulnerable to advanced threats like zero-day exploits and multi-channel Denial-of-Service attacks [112,117]. In response, researchers recommend differential privacy, cryptography, and secure training protocols to strengthen resilience against adversarial manipulation, along with self-consistency sampling to reduce biased outputs [115]. Hybrid repair approaches that balance preservation of correct behavior with targeted fixes [108] and patch optimization strategies to minimize unnecessary edits [67] also provide promising solutions. Additionally, frameworks incorporating Identity and Access Management and detailed audit logs are recommended to ensure traceability [49]. Moreover, resilient multi-agent frameworks, are being developed to enhance the security of Small Language Models against adversarial assaults [94].

RQ4 Finding #4: Mitigating critical security risks in self-healing AI—ranging from adversarial attacks to infrastructure vulnerabilities through advanced defenses like differential privacy, hybrid repair strategies, and rigorous auditing.

4.4.5. Generalization and adaptability across diverse contexts

Another ethical challenge arises from the difficulty of generalizing AI models across heterogeneous datasets and real-world environments [10,27]. Data heterogeneity, such as non-IID distributions across devices, can significantly degrade model performance and reduce fairness [10]. In addition, LLMs often struggle with long prompts, limited attention, or rigid template reliance, which constrains their adaptability to unseen scenarios [44,67]. To address these limitations, proposed solutions include multi-task and transfer learning architectures to better handle diverse contexts [10], as well as segmented prompting and curriculum learning to improve adaptability in large-scale models [43, 67]. Hybrid modeling strategies that adjust thresholds and leverage domain adaptation further strengthen generalization across complex data distributions [43,55].

RQ4 Finding #5: To improve adaptability and handle heterogeneous data, researchers employ multi-task learning, transfer learning, and hybrid modelling strategies.

4.4.6. HITL, Governance, and Accountability

The absence of clear human oversight in AI-driven systems poses ethical risks, as models may operate autonomously without adequate mechanisms for intervention when failures occur [44]. Developers also report distrust in automated repairs, citing concerns that AI-generated patches may diverge from business logic or introduce unintended vulnerabilities [105]. To counter these challenges, several studies emphasize the importance of a HITL approach, where experts validate, correct, or refine automated outputs [76]. This not only improves adaptability to changing conditions but also ensures accountability by keeping humans actively engaged in the decision-making process. Embedding structured feedback loops and manual review mechanisms therefore provides a balance between automation efficiency and ethical responsibility. To address the risk of catastrophic system instability caused by autonomous high-risk actions, recent frameworks introduce an autonomic threshold (α); this mechanism explicitly requires HITL intervention for tasks exceeding a calculated risk level, ensuring safety standards are met [114].

RQ4 Finding #6: To ensure accountability and build trust, researchers advocate for HITL mechanisms that validate and refine AI outcomes.

4.4.7. Standardization and regulation

The lack of standardization in monitoring and digital twin frameworks creates uncertainty in ethical governance, as different implementations may handle data and operational integrity inconsistently [25]. Moreover, compliance with international standards and regulations such as GDPR, PDPL, ISO 8000, and ISO 23247 remains critical to ensuring lawful and ethical use of AI technologies [25,73]. Beyond compliance, the risk of AI weaponization has been highlighted, with calls for global regulatory frameworks to mitigate large-scale harm [58]. To address these challenges, researchers advocate for the integration of standardized monitoring protocols [25], regulation-driven design that embeds compliance into system architectures [73], and international policy efforts to curb the malicious use of AI [58].

RQ4 Finding #7: To ensure compliance and prevent misuse, researchers advocate for standardized protocols and regulation-driven design which incorporates compliance into system architectures.

4.4.8. Operational trade-offs and practical limitations

Operational risks emerge when inaccurate or poorly calibrated AI predictions directly affect maintenance, monitoring, or retraining decisions. For example, incorrect Remaining Useful Life predictions can trigger premature or delayed maintenance actions, undermining trust and safety [116]. Similarly, federated models face a trade-off between lower accuracy and stronger privacy, raising the risk of misleading aggregated insights due to phenomena such as Simpson's Paradox [32], while retraining processes may induce costly overfitting or unintended behavioral shifts [108]. These operational trade-offs extend to vehicular edge computing, where balancing offloading delay against energy consumption often necessitates non-cooperative learning approaches to satisfy privacy constraints [90], and to agentic program repair, where excessive iterations can lead to diminishing returns and a decreased apply rate due to interference [102]. Addressing these multi-faceted challenges requires non-intrusive sensing and regression models that balance accuracy with operational efficiency [116], as well as optimization-based repair strategies that selectively modify neural weights without degrading overall accuracy [108].

RQ4 Finding #8: To balance accuracy with efficiency and privacy, researchers employ non-intrusive sensing and optimization-based repair strategies.

Table 8
Summary of future trends and research directions in AI-driven self-healing.

Category	Trend description	References
Data	Data quality, noise resilience, and cleaning pipelines	[37,75]
	Self-healing Data Pipelines	[49]
	Multimodal observability	[73,96]
	Benchmarks, shared datasets, and reproducibility	[28,51,96]
	Privacy-preserving handling	[30,31,38,45,73]
Model	Agentic AI & GenAI Hyper-automation	[87,114]
	Advanced Architectures	[29,51,90,101]
	Adaptability & Transfer Learning	[10]
	Ontology-enhanced reasoning & Semantic modeling	[41]
	NLP-driven localization & Model-driven approaches	[97]
	LLM/SLM-driven repair, RAG & Cross-language generalization	[8,47,94,98,111]
	Continuous, Online, RL & Adaptive Learning	[28,31,52,118]
	Unsupervised, Semi-supervised & Self-supervised Learning	[37]
	Explainability, Interpretability & Visual Analytics	[51,69,82,89]
Infrastructure	Edge-Cloud Continuum, 5G/6G & Distributed Collaboration	[30,81,119,120]
	Distributed & Modular Digital Twins	[25]
	Scalability, Lightweight Models & Inference Efficiency	[29,38,73]
	End-to-end Autonomous Loops & Parallel Restoration	[82,92,96]
	Sustainability, Green AI & Robotics	[48]
	AI-native DevOps Integration	[43]
Security	Adversarial Resilience, Model Robustness & Secure Management	[26,31,38,118]
	Privacy-preserving Intelligence	[10,31,38,45]
	Blockchain & Distributed Trust Mechanisms	[26,31,45,112,119]
	Vulnerability-aware APR	[8,47]
Governance	HITL, Interactive Feedback & Trust	[45,51,89,99]
	Evaluation Standards, Metrics & Benchmarks	[25,28,69,96]
	Governance, Compliance & Ethical AI	[25,73]
	Equity & Accessibility (Cost-effective AI)	[121]
	Decentralized Coordination	[88]

4.5. RQ5: What are the future trends of automated program repair and intelligent monitoring?

To address RQ5, the emerging directions and open challenges identified across the surveyed literature are analyzed. These future trends are categorized into five key dimensions: data availability and privacy, model advancements, infrastructure scalability, security, and human-centric governance. The following sections detail the specific developments within each area, highlighting the shift toward more resilient, autonomous, and trustworthy self-healing systems. A summary of these trends and their corresponding primary references is provided in Table 8

4.5.1. Data trends

Future research on self-healing systems increasingly prioritizes the quality and accessibility of data. Reliable mechanisms depend on signals that are resilient to noise and consistently available, underscoring the need for robust preprocessing and cleaning pipelines [37,75]. A notable emerging trend is the development of self-healing data pipelines, where systems combine data engineering with AI automation to continuously retrain models and refine policies based on feedback loops, ensuring long-term relevance [49]. Concurrently, there is a shift toward richer, multimodal observability — combining sensor readings, visual inputs, and metadata — to improve contextual awareness [73,96]. A persistent barrier remains the scarcity of large-scale, realistic datasets; consequently, the community is calling for shared benchmarks that capture real-world variability [28,51,96]. Finally, as data volume grows, privacy-preserving techniques such as federated learning are becoming essential for enabling distributed learning without compromising confidentiality [30,31,38,45,73].

RQ5 Finding #1: Future research emphasizes high-quality multimodal data and shared benchmarks, while prioritizing privacy through federated learning and compliance.

4.5.2. Model trends

The reviewed literature highlights a significant progression toward learning-based approaches that enhance the adaptability, robustness, and autonomy of intelligent monitoring systems and APR. A major trend characterizing this evolution is the shift from reactive firefighting to proactive Agentic AI and GenAI-driven hyper-automation, where autonomous agents not only detect faults but act as strategists to prevent them [87,114].

To support this shift, future research emphasizes the adoption of advanced AI architectures capable of capturing intricate structural and temporal connections. This includes the use of graph-based models and topology-aware learning, which are crucial for overcoming partial observability in distributed systems [29,51,90,101]. Complementary techniques such as hybrid neural methods, physics-informed AI, and RL are expected to drive adaptive anomaly response [29,51,101]. In this context, future methods aim to apply RL to enable systems to learn from experience and improve decision-making continuously through feedback [118]. Furthermore, transfer learning and multi-task learning are identified as key strategies for handling new situations with limited data, contributing to more flexible and generalizable models [10]. In parallel, personalized federated learning and strategies for non-IID data handling are anticipated to play a central role in preserving privacy while improving global model accuracy in large-scale deployments [10].

A significant evolution is observed in the domain of Generative AI, specifically the integration of LLMs and SLMs. While LLMs facilitate complex tasks—such as Agentic Translation for program repair, cross-language generalization, and Retrieval-Augmented Generation (RAG) for repository-level context [8,47,98,111]—there is a growing focus on SLMs. These smaller models are essential for enabling secure, resilient execution in resource-constrained edge environments [94]. To address the dynamic nature of these environments, continuous and online learning are highlighted as mechanisms to adapt to concept drift and evolving log patterns without full retraining [28,31,52], while unsupervised and self-supervised learning help overcome data scarcity [37].

Beyond pure learning approaches, ontology-enhanced reasoning and semantic knowledge modeling are increasingly emphasized as

critical enablers. These techniques complement data-driven methods by leveraging rich semantic relationships between software artifacts to ensure precise bug localization and adaptive repair strategies [41]. Similarly, model-driven approaches integrating NLP are emerging as key components for automated self-healing, particularly in multi-language and deep-learning-specific contexts [97].

Finally, to maintain automation and ensure operator trust, future models must incorporate robust XAI and interpretability mechanisms. These include feature-level anomaly explanations, visual analytics for decision support, and DL models that provide insights into fix generation [51,69,82]. This necessitates balancing fully autonomous loops with hybrid HITL collaboration models, ensuring transparency and accountability in decision-making processes [89].

RQ5 Finding #2: Future trends prioritize the shift toward proactive Agentic AI and topology-aware architectures, leveraging the synergy of LLMs and continuous learning to drive robust, autonomous repair. These advancements are complemented by ontology-enhanced reasoning and XAI to ensure precision, transparency, and operator trust in increasingly complex systems.

4.5.3. Infrastructure trends

Future infrastructure directions center on creating scalable, distributed, and low-latency systems capable of supporting real-time monitoring and autonomous self-healing. A dominant trend is the evolution of the Edge–Cloud Continuum, where computation is dynamically distributed across edge, fog, and cloud layers to balance latency, resource constraints, and fault tolerance [30,81,119]. This continuum is expanding to incorporate advanced connectivity, such as 5G/6G and satellite links, to ensure failover redundancy for critical operations [30,119]. Similarly, in [120] introduces a multi-agent framework for autonomous distributed restoration in smart distribution networks, specifically addressing the flexibility challenges posed by renewable energy uncertainty.

In this distributed landscape, Digital Twins are expected to evolve into modular architectures that enable real-time simulation, diagnosis, and control across large industrial deployments [25]. Complementing this, recovery strategies are shifting from sequential methods to parallel autonomous restoration, allowing simultaneous recovery across multiple zones to drastically reduce outage durations [25,92].

Scalability and performance efficiency remain critical challenges, driving research into lightweight models, optimized inference pipelines, and adaptive deployment strategies suited for massive environments [29,38,73]. This focus aligns with the emerging priority of sustainability and "Green AI", which seeks to reduce energy consumption and waste, including the integration of robotics for physical autonomous maintenance [48].

Finally, a major trend is the deep integration of automated remediation into end-to-end operational workflows, forming continuous closed loops capable of autonomous action [82,96]. The operationalization of AI within DevOps — often framed as LLMOps or AI-native DevOps — aims to embed these self-healing and program repair capabilities directly into CI/CD pipelines and infrastructure-as-code processes [43].

RQ5 Finding #3: Future infrastructure trends prioritize scalable Edge–Cloud continuums and modular Digital Twins to support real-time self-healing, while integrating Green AI and parallel restoration strategies to ensure efficiency and sustainability in autonomous operational workflows.

4.5.4. Security trends

Security and trustworthiness are becoming central to the future of self-healing systems, particularly in distributed and sensitive environments. Several studies call for more resilient detection mechanisms that can withstand adversarial conditions, model poisoning, and evolving cyber threats [26,31,38]. Privacy-preserving intelligence is equally important, with federated learning, secure multiparty computation, differential privacy, and encrypted model updates highlighted as necessary tools for enabling collaborative learning without exposing raw data [10,31,38,45]. A parallel trend involves leveraging blockchain and distributed trust mechanisms to ensure tamper-proof logging, visibility, and traceability of autonomous decisions [26,45,112,119], as well as to enhance the security and integrity of management processes [118]. Complementing these directions, vulnerability-aware APR is gaining traction, expanding the scope of APR from functional defects to systematic security weaknesses [8,47].

RQ5 Finding #4: Future security focuses on adversarial resilience, privacy preservation via federated learning, blockchain-backed integrity, and automated vulnerability repair.

4.5.5. Human factors, evaluation, and governance trends

Beyond technical advances, future work highlights the importance of human factors, evaluation practices, and governance frameworks necessary to mature self-healing and automated repair technologies. In [45,51,89,99], the authors emphasize that humans will remain part of the loop, whether through feedback-driven learning, operator-informed anomaly annotation, or interactive repair validation in development environments. The need for comprehensive benchmarks and standardized evaluation methodologies also emerges as a significant trend. Researchers call for long-term datasets, realistic fault scenarios, and consistent metrics that enable reproducibility and fair comparison across monitoring and APR systems [25,28,69,96]. There is also a rising demand for cost-effective equity, ensuring that advanced self-healing solutions are accessible to less economically developed regions through lightweight or cloud-shared frameworks [121]. Governance and compliance concerns further shape the research agenda, particularly in domains where sensitive data, safety, or regulation play a critical role. Works addressing GDPR-compliant monitoring, ethical AI, and standardized digital-twin frameworks illustrate the growing requirement to align technical solutions with policy and operational constraints [25,73]. Recent work also identifies decentralized, collaborative self-healing as a key governance trend, stressing the need for frameworks that manage coordination and decision-making across distributed monitors without centralized control [88].

RQ5 Finding #5: Future advancements prioritize HITL collaboration and the establishment of standardized benchmarks to ensure the reproducibility and maturity of self-healing systems. Concurrently, the focus is expanding to include equitable accessibility and robust governance frameworks—addressing data privacy, ethical compliance, and decentralized coordination in distributed environments.

5. Related works

In [122], the authors highlight the increase in system failures due to external attacks or internal process malfunctions. To mitigate them, they perform a critical analysis on Machine Learning-driven self-healing theories and methods for cyber–physical systems. They acknowledge the existing gaps for real-world adoption, but forecast an increase in the development and adoption of these techniques. Similarly, Adeniyi et al. [125] conduct a systematic literature review focused on proactive self-healing within Mobile Edge Computing (MEC). They propose a taxonomy categorized by objectives, attributes, and approaches — such

Table 9
Summary of related works conducted in areas covered by this research and their comparison with our SLR.

Study	Type	Methodology	Period	Included studies	Scope	Features
Johnphill et al. [122]	Review	CA[a]	Up to 2022[b]	–[c]	Fault diagnosis, Fault isolation, System recovery	ML-based remediation in CPS[d].
Liang et al. [123]	Review	General review	Up to 2023[b]	–[c]	System stability, Autonomous recovery	Control Theory, Industrial Automated Systems, Fault-tolerant Control.
Feng et al. [121]	Review	General review	Up to 2025[b]	–[c]	Fault diagnosis, Fault isolation, System recovery	Multi-Agent Systems, RL
Souza et al. [124]	Survey	Lit. analysis	2017–2023	49	Software maintenance, Preventive and corrective maintenance.	Maintenance patterns in IoT/Edge.
Adeniyi et. al. [125]	SLR	PRISMA	2018–2021	116	Proactive Self-healing	MEC, Deep Learning, QoS[e]
Tawfeeg et. al. [126]	SLR	Standard SLR	2011–2021	64	Reactive Self-healing	Cloud Computing, VM Migration
This study	SLR	Kitchenham and PRISMA	2021–2026	99	End-to-end autonomous Remediation, Agentic self-healing, Self-adaptive APR ecosystem	Agentic AI, LLM, Edge-Cloud Continuum, Ethical AI Governance.

[a] Critical Analysis.
[b] Time frame not explicitly stated.
[c] Not Determined.
[d] Cyber-Physical Systems.
[e] Quality of Service.

as SDN and Kubernetes — while identifying critical gaps in MEC resource management and security. Feng et al. [121] focus on subway power supply systems as a critical component of urban rail transit infrastructure, and highlight the challenges of adopting self-healing to these technologies. In this review, they study the integration of multi-agent systems and AI and identify the benefits of these approaches on subway power systems. These benefits include the capacity to handle complex, multi-fault scenario that overwhelm traditional control methods, and the possibility to continuously learn and adapt to new fault patterns.

The authors explore the wide range of applications in which self-healing technologies can be utilized is analyzed in [123]. They propose the definition, theoretical framework, and characteristics of self-healing control. They also summarize the development tendency of self-healing control and lay the foundation for the theoretical exploration of self-healing control. In [124], the authors elaborate on the challenges presented by IoT technologies in conjunction with Cloud and Edge Computing paradigms. They highlight the need for efficient maintenance strategies for the managing these complex infrastructures. The provide a taxonomy structure to facilitate the understating of the research landscape and provide open challenges that pave the way for promising research directions. Complementing these maintenance strategies, Tawfeeg et al. [126] provide an SLR on the integration of dynamic load balancing and reactive fault tolerance in cloud computing. Their work emphasizes the relationship between these two techniques to ensure high service availability, noting that while many researchers utilize nature-inspired methods for task scheduling, there remains a need for more standardized evaluation metrics across the field.

Table 9 provides a comparative analysis between the findings of this SLR and existing literature. Unlike previous reviews that often lack a structured methodology or focus on specific domains, such as industrial control systems [123] or IoT maintenance [124], this manuscript follows a rigorous framework combining Kitchenham's guidelines [13,14] and the PRISMA 2020 [15] methodology to ensure transparency and objectivity. While recent studies by Adeniyi et al. [125] and Tawfeeg et al. [126] focus on proactive and reactive mechanisms respectively, our study covers a more recent time frame (2021–2026) and broadens the scope to include agentic self-healing and the self-adaptive APR ecosystem. Ultimately, this work distinguishes itself by integrating modern paradigms like LLMs and ethical AI governance within the Edge–Cloud continuum, providing a future-oriented roadmap for autonomous remediation.

6. Threats to validity

Following the guidelines of Ampatzoglou et al. [127], the relevant threats to validity for this study are categorized into three types: study selection validity, data validity and research validity. Most of the threats listed by Ampatzoglou et al. are addressed in the quality assurance plan presented in Table 4. Furthermore, the PRISMA 2020 statement [15] and the systematic literature review guidelines for software engineering by Kitchenham and Brereton [13,14] were followed to ensure the application of state-of-the-art methods throughout the study.

6.1. Study selection validity

The systematic process illustrated in Section 3 addresses most of the threats to validity listed by Ampatzoglou et al. [127] in this category. Using Scopus as the main bibliographic database because of its broad coverage of important publishers like IEEE, ACM, Springer, and Elsevier, among others, a number of mitigation strategies were specifically used to guarantee the discovery of pertinent works. A multi-dimensional search taxonomy and pilot searches were used to improve the search procedure. After processing 2930 original records to ultimately choose 99 main studies, the application of inclusion and exclusion criteria is regarded as rigorous. Although non-English-language sources were not taken into account, this is not seen as a major drawback considering the worldwide scope of Edge–Cloud Continuum and AI-driven self-healing research.

6.2. Data validity

The final list of 99 papers covers a wide range of esteemed academic settings and is quite pertinent to the research concerns. Because of this, it is adequate and representative for a thorough synthesis of the field. A rigorous evaluation was carried out using seven different quality metrics (M1–M7), as indicated in Table 4, to guarantee the quality of the evidence. Papers with strong research foundations and transparent methodology were given priority. Data extraction was done methodically using a standardized form in accordance with the quality assurance strategy, and researchers conducted random quality checks to ensure objective data extraction and minimize subjectivity in understanding intricate AI structures.

6.3. Research validity

The research process has been documented extensively, including the research protocol reported in Section 3. The chosen SLR method is deemed suitable for answering the research questions, as the explicit goal was to synthesize the state-of-the-art and implementation challenges regarding AI-driven self-healing. Related work is used throughout the report, particularly in Section 4, where findings are discussed in relation to the IBM MAPE-K model [128] and other existing proposals. The study is believed to be generalizable within its domain, as it captures a broad understanding of current literature across the entire Computing Continuum, from edge devices to cloud infrastructures.

7. Conclusion and future works

The overarching goal of this SLR is to offer insights into the state of autonomous self-healing systems across the cloud–edge continuum. An in-depth search of the scientific literature was conducted on the search terms described in Fig. 1 and the discussion is elaborated based on the research questions outlined in Table 1. The analysis confirms that the management of modern distributed architectures — characterized by microservices, serverless deployments, and non-deterministic faults — can no longer rely solely on traditional centralized operations. Instead, the convergence of AIOps with LLMs and Agentic pipelines is establishing a new baseline for engineering software that is intrinsically resilient. While current approaches demonstrate significant capabilities in causal discovery and code generation, the transition from observational monitoring to actionable, autonomous repair remains hindered by challenges in trust, validation, and architectural complexity.

To advance the state of the art, future research must prioritize the rigorous software testing and verification & validation of AI-generated remedial actions. A critical barrier identified in this study is the "validation gap" where LLM-generated patches lack deterministic guarantees. Future work should therefore focus on developing novel quality assurance frameworks that integrate formal verification methods and runtime monitoring directly into the repair loop. This aligns with the need for robust software quality and metrics that can quantitatively assess the safety and correctness of autonomous agents operating in production environments, ensuring that self-healing mechanisms do not introduce regressions or security vulnerabilities.

Furthermore, the engineering of these complex systems necessitates a re-evaluation of software architecture and design patterns. The distinct constraints of the continuum demand a shift towards decentralized, multi-agent architectures where intelligence is distributed between the edge and the cloud. Research into design for repairability principles will be essential, enabling developers to architect systems that are observable and controllable by autonomous agents by design. This evolution also extends to software management and processes, where the integration of HITL governance models is paramount. As AIOps matures into LLMOps, future contributions must address the seamless incorporation of these autonomous workflows into standard DevOps lifecycles, balancing automation with the necessary ethical compliance and explainability.

Finally, looking toward future research avenues, the rapid integration of Agentic AI into self-healing workflows represents a transformative trend that necessitates comprehensive empirical studies to bridge the widening gap between theoretical algorithms and industrial application. This manuscript categorizes the associated obstacles into five key dimensions: data availability and privacy, model advancements, infrastructure scalability, security, and human-centric governance. To navigate these complexities, there is a pressing need for the community to prioritize the creation of standardized, high-quality benchmarks and real-world datasets that allow for the comparative evaluation of self-healing frameworks. This is particularly urgent as the upsurge in LLMs continues to revolutionize automated repair, requiring robust validation for cross-language generalization and vulnerability remediation. By addressing these challenges, ranging from the architectural distribution of intelligence to the rigorous governance of autonomous agents, researchers can significantly improve the methods used to engineer and manage the next generation of resilient software systems, ultimately realizing the vision of truly autonomous operations.

CRediT authorship contribution statement

Lander Bonilla: Writing – review & editing, Writing – original draft, Visualization, Investigation, Formal analysis, Conceptualization. **Josu Diaz-de-Arcaya:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization. **Jon Aguirre-Usandizaga:** Writing – review & editing, Conceptualization. **Aitor Almeida:** Writing – review & editing, Supervision, Funding acquisition.

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.

Acknowledgments

This work was supported by NexusForum.EU project, that is a Coordination & Support Action co-funded by the European Union's Horizon Europe research and innovation programme under Grant Agreement 101135632 and by the Swiss State Secretariat for Education, Research, and Innovation (SERI). The objective of NexusForum.EU is to boost the consolidation of the European Computing Continuum ecosystem, as well as to provide a forward-looking and bold vision in new areas and directions that have not been explored so far.

Data availability

Data will be made available on request.

References

[1] M. Parashar, Autonomic computing rebooted: Taming the computing continuum, ACM Trans. Auton. Adapt. Syst. (2025).
[2] Q.M. Ashraf, M. Tahir, M.H. Habaebi, J. Isoaho, Toward autonomic internet of things: Recent advances, evaluation criteria, and future research directions, IEEE Internet Things J. 10 (16) (2023) 14725–14748.
[3] C. Savaglio, G. Fortino, Autonomic and cognitive architectures for the internet of things, in: International Conference on Internet and Distributed Computing Systems, Springer, 2015, pp. 39–47.
[4] L. Zhang, T. Jia, M. Jia, Y. Wu, A. Liu, Y. Yang, Z. Wu, X. Hu, P. Yu, Y. Li, A survey of aiops in the era of large language models, ACM Comput. Surv. (2025).
[5] T. Szandała, Aiops for reliability: Evaluating large language models for automated root cause analysis in chaos engineering, in: International Conference on Computational Science, Springer, 2025, pp. 323–336.
[6] C.-M. Lin, C. Chang, W.-Y. Wang, K.-D. Wang, W.-C. Peng, Root cause analysis in microservice using neural granger causal discovery, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 206–213.
[7] L. Pham, H. Ha, H. Zhang, Root cause analysis for microservice system based on causal inference: How far are we? in: Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering, 2024, pp. 706–715.
[8] F. Zubair, M. Al-Hitmi, C. Catal, The use of large language models for program repair, Comput. Stand. Interfaces 93 (2025) 103951.
[9] B. Yang, Z. Cai, F. Liu, B. Le, L. Zhang, T.F. Bissyandé, Y. Liu, H. Tian, A survey of llm-based automated program repair: Taxonomies, design paradigms, and applications, 2025, arXiv preprint arXiv:2506.23749.
[10] P. Sharma, S.K. Sharma, D. Dani, Edge-assisted federated learning for anomaly detection in diverse iot network, Int. J. Inf. Technol. 17 (5) (2025) 3035–3045.
[11] V. Kansal, S.O. Husain, R. Kumar, N. Chinthamu, M. Kumar, S. Aswini, Edge computing enabled anomaly detection in iot environments using federated learning, in: 2024 International Conference on Communication Computer Sciences and Engineering, IC3SE, IEEE, 2024, pp. 1622–1627.
[12] E. Birihanu, I. Lendák, Explainable correlation-based anomaly detection for industrial control systems, Front. Artif. Intell. 7 (2025) 1508821.

[13] S. Keele, et al., Guidelines for Performing Systematic Literature Reviews in Software Engineering, Tech. rep., Technical report, ver. 2.3 ebse technical report, ebse, 2007.

[14] B. Kitchenham, P. Brereton, A systematic review of systematic review process research in software engineering, Inf. Softw. Technol. 55 (12) (2013) 2049–2075.

[15] M.J. Page, J.E. McKenzie, P.M. Bossuyt, I. Boutron, T.C. Hoffmann, C.D. Mulrow, L. Shamseer, J.M. Tetzlaff, E.A. Akl, S.E. Brennan, et al., The prisma 2020 statement: an updated guideline for reporting systematic reviews, Bmj 372 (2021).

[16] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, in: 12th International Conference on Evaluation and Assessment in Software Engineering, EASE, BCS Learning & Development, 2008, pp. 1–10.

[17] J. Diaz-De-Arcaya, A.I. Torre-Bastida, G. Zarate, R. Minon, A. Almeida, A joint study of the challenges, opportunities, and roadmap of mlops and aiops: A systematic survey, ACM Comput. Surv. 56 (4) (2023) 1–30.

[18] B.V. Elsevier, Elsevier developer portal, 2026, (Accessed: 14 April 2026) https://dev.elsevier.com/.

[19] Springer Nature, Springer nature API portal, 2026, (Accessed: 14 April 2026) https://dev.springernature.com/restfuloperations.

[20] IEEE, IEEE X plore API portal, 2026, (Accessed: 14 April 2026) https://developer.ieee.org/.

[21] R. Verdecchia, J. Sallou, L. Cruz, A systematic review of green ai, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 13 (4) (2023) e1507.

[22] J.M. Zhang, M. Harman, L. Ma, Y. Liu, Machine learning testing: Survey, landscapes and horizons, IEEE Trans. Softw. Eng. 48 (1) (2020) 1–36.

[23] Q. Mahood, D. Van Eerd, E. Irvin, Searching for grey literature for systematic reviews: challenges and benefits, Res. Synth. Methods 5 (3) (2014) 221–234.

[24] L. Bonilla Viana, J. Diaz de Arcaya, J. Aguirre Usandizaga, A. Almeida, AI-Driven Self-Healing Across the Edge-Cloud Continuum: A Systematic Literature Review (Replication Package), Zenodo, 2026, http://dx.doi.org/10.5281/zenodo.20024411, (Accessed: 04 May 2026).

[25] I. Abdullahi, S. Longo, M. Samie, Towards a distributed digital twin framework for predictive maintenance in industrial internet of things (iiot), Sensors 24 (8) (2024) 2663.

[26] A.B. Dorothy, B. Madhavidevi, B. Nachiappan, G. Manikandan, P.K. Patjoshi, M. Sindhuja, AI-driven threat intelligence in cloud computing detecting and responding to cyber attacks, in: 2024 International Conference on Intelligent Algorithms for Computational Intelligence Systems, IACIS, IEEE, 2024, pp. 1–6.

[27] R. Rouf, M. Rasolroveicy, M. Litoiu, S. Nagar, P. Mohapatra, P. Gupta, I. Watts, Instantops: A joint approach to system failure prediction and root cause identification in microserivces cloud-native applications, in: Proceedings of the 15th ACM/SPEC International Conference on Performance Engineering, 2024, pp. 119–129.

[28] W. Sus, P. Nawrocki, Signature-based adaptive cloud resource usage prediction using machine learning and anomaly detection, J. Grid Comput. 22 (2) (2024) 46.

[29] Y. Chang, J. Chen, R. Su, J. Xie, A. Li, Two-phase dual-adversarial agents with multivariate information for unsupervised anomaly detection of iiot-edge devices, IEEE Internet Things J. 11 (13) (2024) 23577–23591.

[30] S.V. Haldikar, O.F.M.A. Kader, R.K. Yekollu, Edge computing and federated learning for real-time anomaly detection in industrial internet of things (iiot), in: 2024 International Conference on Inventive Computation Technologies, ICICT, IEEE, 2024, pp. 1699–1703.

[31] A.V. Nagarjun, S. Rajkumar, Design of an anomaly detection framework for delay and privacy-aware blockchain-based cloud deployments, IEEE Access 12 (2024) 84843–84862.

[32] A. Bemani, N. Björsell, Aggregation strategy on federated machine learning algorithm for collaborative predictive maintenance, Sensors 22 (16) (2022) 6252.

[33] A.M. El-Shamy, N.A. El-Fishawy, G. Attiya, M.A. Mohamed, Anomaly detection and bottleneck identification of the distributed application in cloud data center using software–defined networking, Egypt. Informatics J. 22 (4) (2021) 417–432.

[34] B. Lin, S. Wang, M. Wen, X. Mao, Context-aware code change embedding for better patch correctness assessment, ACM Trans. Softw. Eng. Methodol. (TOSEM) 31 (3) (2022) 1–29.

[35] M. Heyden, J. Wilwer, E. Fouché, S. Thoma, S. Matthiesen, T. Gwosch, Tandem outlier detectors for decentralized data, in: Proceedings of the 34th International Conference on Scientific and Statistical Database Management, 2022, pp. 1–4.

[36] Z. Wang, J. Tian, H. Fang, L. Chen, J. Qin, Lightlog: A lightweight temporal convolutional network for log anomaly detection on the edge, Comput. Netw. 203 (2022) 108616.

[37] Y. Liu, W. Yu, T. Dillon, W. Rahayu, M. Li, Empowering iot predictive maintenance solutions with AI: A distributed system for manufacturing plant-wide monitoring, IEEE Trans. Ind. Informatics 18 (2) (2021) 1345–1354.

[38] M.S. Reddy, K. Chatterjee, M. Raju, S.S. Kumar, T.A.V. Reddy, M.N. Thara, Fedanom: Strengthening cloud network security with federated anomaly detection, in: 2024 IEEE International Conference on Information Technology Electronics and Intelligent Communication Systems, ICITEICS, IEEE, 2024, pp. 1–7.

[39] R. Siraskar, S. Kumar, S. Patil, A. Bongale, K. Kotecha, Application of the nadaraya-watson estimator based attention mechanism to the field of predictive maintenance, MethodsX 12 (2024) 102754.

[40] J. Zhang, C. Wang, A. Li, W. Wang, T. Li, Y. Liu, Vuladvisor: Natural language suggestion generation for software vulnerability repair, in: Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering, 2024, pp. 1932–1944.

[41] A.S.D. Silva, R.E. Garcia, L.C. Botega, Bug localization model in source code using ontologies, IEEE Access 11 (2023) 98542–98557.

[42] P. Hu, L. Pan, Y. Chen, Z. Fang, L. Huang, Effective multi-user delay-constrained scheduling with deep recurrent reinforcement learning, in: Proceedings of the Twenty-Third International Symposium on Theory Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, 2022, pp. 1–10.

[43] A. Zirak, H. Hemmati, Improving automated program repair with domain adaptation, ACM Trans. Softw. Eng. Methodol. 33 (3) (2024) 1–43.

[44] Y. Wei, C.S. Xia, L. Zhang, Copiloting the copilots: Fusing large language models with completion engines for automated program repair, in: Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2023, pp. 172–184.

[45] S. Shaikh, M. Jammal, Survey of fault management techniques for edge-enabled distributed metaverse applications, Comput. Netw. 254 (2024) 110803.

[46] J. Alonso, L. Orue-Echevarria, E. Osaba, J.L. Lobo, I. Martinez, J.D. de Arcaya, I. Etxaniz, Optimization and prediction techniques for self-healing and self-learning applications in a trustworthy cloud continuum, Information 12 (8) (2021) 308.

[47] K. Huang, X. Meng, J. Zhang, Y. Liu, W. Wang, S. Li, Y. Zhang, An empirical study on fine-tuning large language models of code for automated program repair, in: 2023 38th IEEE/ACM International Conference on Automated Software Engineering, ASE, IEEE, 2023, pp. 1162–1174.

[48] C. Wang, K. Liu, Y. Yuan, S. Peng, G. Li, Joint trajectory and offloading optimization in uav-assisted mec via federated multi-agent reinforcement learning and potential fields, Comput. Netw. (2025) 111681.

[49] S. Chippagiri, K. Alang, A. Gumber, S.G. Thomas, Autonomous data quality monitoring with ai agents: Integrating ml with cloud warehouses and data lakes, in: 2025 International Conference on Computing Technologies & Data Communication, ICCTDC, IEEE, 2025, pp. 1–7.

[50] S. Meng, L. Luo, X. Qiu, Y. Dai, Service-oriented reliability modeling and autonomous optimization of reliability for public cloud computing systems, IEEE Trans. Reliab. 71 (2) (2022) 527–538.

[51] S. Rigas, P. Tzouveli, S. Kollias, An end-to-end deep learning framework for fault detection in marine machinery, Sensors 24 (16) (2024) 5310.

[52] Y. Duan, K. Xue, H. Sun, H. Bao, Y. Wei, Z. You, Y. Zhang, X. Jiang, S. Yang, J. Chen, Logedl: Log anomaly detection via evidential deep learning, Appl. Sci. 14 (16) (2024) 7055.

[53] K. Mala, H.S. Annapurna, Cloud network traffic classification and intrusion detection system using deep learning, in: 2023 International Conference on Integrated Intelligence and Communication Systems, ICIICS, IEEE, 2023, pp. 1–8.

[54] Q. Zhang, C. Fang, W. Sun, Y. Liu, T. He, X. Hao, Z. Chen, Appt: Boosting automated patch correctness prediction via fine-tuning pre-trained models, IEEE Trans. Softw. Eng. 50 (3) (2024) 474–494.

[55] A.H. Mohammadkhani, C. Tantithamthavorn, H. Hemmatif, Explaining transformer-based code models: What do they learn? when they do not work? in: 2023 IEEE 23rd International Working Conference on Source Code Analysis and Manipulation, SCAM, IEEE, 2023, pp. 96–106.

[56] C.S. Xia, Y. Wei, L. Zhang, Automated program repair in the era of large pre-trained language models, in: 2023 IEEE/ACM 45th International Conference on Software Engineering, ICSE, IEEE, 2023, pp. 1482–1494.

[57] Y. Duan, H. Bao, G. Bai, Y. Wei, K. Xue, Z. You, Y. Zhang, B. Liu, J. Chen, S. Wang, Learning to diagnose: Meta-learning for efficient adaptation in few-shot aiops scenarios, Electronics 13 (11) (2024) 2102.

[58] S. Ali, J. Wang, V.C.M. Leung, AI-driven fusion with cybersecurity: Exploring current trends, advanced techniques, future directions, and policy implications for evolving paradigms–a comprehensive review, Inf. Fusion (2025) 102922.

[59] J. Baek, G. Kaddoum, Online partial offloading and task scheduling in SDN-fog networks with deep recurrent reinforcement learning, IEEE Internet Things J. 9 (13) (2021) 11578–11589.

[60] X. Li, J. Wu, X. Ling, T. Luo, Y. Wu, Automatic program repair via learning edits on sequence code property graph, in: 2023 IEEE 29th International Conference on Parallel and Distributed Systems, ICPADS, IEEE, 2023, pp. 563–570.

[61] M. Shetty, Y. Chen, G. Somashekar, M. Ma, Y. Simmhan, X. Zhang, J. Mace, D. Vandevoorde, P. Las-Casas, S.M. Gupta, Building ai agents for autonomous clouds: Challenges and design principles, in: Proceedings of the 2024 ACM Symposium on Cloud Computing, 2024, pp. 99–110.

[62] S. Omari, K. Basnet, M. Wardat, Investigating large language models capabilities for automatic code repair in python, Clust. Comput. 27 (8) (2024) 10717–10731.

[63] I. Ait Abdelmoula, S.I. Kaitouni, N. Lamrini, M. Jbene, A. Ghennioui, A. Mehdary, M. El Aroussi, Towards a sustainable edge computing framework for condition monitoring in decentralized photovoltaic systems, Heliyon 9 (11) (2023).

[64] G.H. Lee, J. Han, An edge-based intelligent iot control system: Achieving energy efficiency with secure real-time incident detection, J. Netw. Syst. Manage. 33 (1) (2025) 13.

[65] S. Jo, J.-Y. Oh, Y.T. Yoon, Y.G. Jin, Self-healing radial distribution network reconfiguration based on deep reinforcement learning, Results Eng. 22 (2024) 102026.

[66] I. Bouzenia, P. Devanbu, M. Pradel, Repairagent: An autonomous, llm-based agent for program repair, 2024, arXiv preprint arXiv:2403.17134.

[67] B. Yang, H. Tian, W. Pian, H. Yu, H. Wang, J. Klein, S. Jin T.F. Bissyandé, Cref: An llm-based conversational software repair framework for programming tutors, in: Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis, 2024, pp. 882–894.

[68] M. Sintaha, N. Nashid, A. Mesbah, Katana: Dual slicing based context for learning bug fixes, ACM Trans. Softw. Eng. Methodol. 32 (4) (2023) 1–27.

[69] W. Zhong, C. Li, J. Ge, B. Luo, Neural program repair: Systems, challenges and solutions, in: Proceedings of the 13th Asia-Pacific Symposium on Internetware, 2022, pp. 96–106.

[70] Q. Zhu, Z. Sun, Y. an Xiao, W. Zhang, K. Yuan, Y. Xiong, L. Zhang, A syntax-guided edit decoder for neural program repair, in: Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2021, pp. 341–353.

[71] K. Dhatchayani, R. Shubavathy, G. Reshma, D. Sundar, D. Vezhaventhan, et al., Optimizing smart city infrastructure using 5 g edge ai with adaptive multi-agent reinforcement learning, in: 2025 International Conference on Visual Analytics and Data Visualization, ICVADV, IEEE, 2025, pp. 1286–1292.

[72] H. Tian, X. Tang, A. Habib, S. Wang, K. Liu, X. Xia, J. Klein, T.F. Bissyandé, Is this change the answer to that problem? correlating descriptions of bug and code changes for evaluating patch correctness, in: Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering, 2022, pp. 1–13.

[73] D. Tsiktsiris, A. Lalas, M. Dasygenis, K. Votis, Enhancing the safety of autonomous vehicles: Semi-supervised anomaly detection with overhead fisheye perspective, IEEE Access 12 (2024) 68905–68915.

[74] H. Tian, Y. Li, W. Pian, A.K. Kabore, K. Liu, A. Habib, J. Klein, T.F. Bissyandé, Predicting patch correctness based on the similarity of failing test cases, ACM Trans. Softw. Eng. Methodol. (TOSEM) 31 (4) (2022) 1–30.

[75] Y. Ledmaoui, A.E. Maghraoui, M.E. Aroussi, R. Saadane, A. Chehri, A. Chebak, PV solar anomaly detection using low-cost data logger and ANN algorithm, TELKOMNIKA (Telecommunication Comput. Electron. Control) 23 (1) (2024) 258–266.

[76] C. Geethal, M. Böhme, V.-T. Pham, Human-in-the-loop automatic program repair, IEEE Trans. Softw. Eng. 49 (10) (2023) 4526–4549.

[77] W. Wang, Y. Wang, S. Hoi, S. Joty, Towards low-resource automatic program repair with meta-learning and pretrained language models, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 6954–6968.

[78] F. Winkel, O. Wallscheid, P. Scholz, J. Böcker, Pseudolabeling machine learning algorithm for predictive maintenance of relays, IEEE Open J. Ind. Electron. Soc. 4 (2023) 463–475.

[79] R. Widyasari, G.A.A. Prana, S.A. Haryono, Y. Tian, H.N. Zachiary, D. Lo, Xai4fl: Enhancing spectrum-based fault localization with explainable artificial intelligence, in: Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension, 2022, pp. 499–510.

[80] S. Caleb, G. Padmapriya, T. Nandhini, R. Latha, et al., Revolutionizing fault detection in self-healing network via multi-serial cascaded and adaptive network, Knowl.-Based Syst. 309 (2025) 112732.

[81] X. Zhao, P. Wang, A deployable edge computing solution for machine condition monitoring, in: 2024 IEEE International Instrumentation and Measurement Technology Conference, I2MTC, IEEE, 2024, pp. 1–6.

[82] Y. Song, R. Xin, P. Chen, R. Zhang, J. Chen, Z. Zhao, Identifying performance anomalies in fluctuating cloud environments: A robust correlative-gnn-based explainable approach, Future Gener. Comput. Syst. 145 (2023) 77–86.

[83] O.T. Basaran, F. Dressler, Xainomaly: Explainable, interpretable and trustworthy ai for xurllc in 6 g open-ran, in: 2024 3rd International Conference on 6G Networking (6GNet), IEEE, 2024, pp. 93–101.

[84] P. Singh, M. Saman Azari, F. Vitale, F. Flammini, N. Mazzocca, M. Caporuscio, J. Thornadtsson, Using log analytics and process mining to enable self-healing in the internet of things, Environ. Syst. Decis. 42 (2) (2022) 234–250.

[85] Y. Cohen, G. Singer, A smart process controller framework for industry 4.0 settings, J. Intell. Manuf. 32 (7) (2021) 1975–1995.

[86] X. Huang, C. Xiang, H. Li, P. He, Sbuglocater: Bug localization based on deep matching and information retrieval, Math. Probl. Eng. 2022 (1) (2022) 3987981.

[87] R.D. Zota, C. Bărbulescu, R. Constantinescu, A practical approach to defining a framework for developing an agentic aiops system, Electronics 14 (9) (2025) 1775.

[88] M. Sanabria, I. Dusaric, N. Cardozo, Learning recovery strategies for dynamic self-healing in reactive systems, in: Proceedings of the 19th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, 2024, pp. 133–142.

[89] A. Sheth, A. Achanta, P. Matam, A. Patel, P. Sharma, N.V.P. Janapareddy, B. Patil, V. Gudur, Ai driven self-healing cybersecurity systems with agentic ai for adaptive threat response and resilience, in: 2025 IEEE Cloud Summit, IEEE, 2025, pp. 147–153.

[90] Y. Lin, L. Xiao, Y. Tao, Y. Zhang, F. Shu, J. Li, Multi-agent computing-energy-efficiency optimization in vehicular edge computing: Non-cooperative versus cooperative solutions, IEEE Trans. Wirel. Commun. (2025).

[91] N. Tiwari, Agentic ai-driven real-time inventory management using distributed cloud architectures and machine learning, in: 2025 International Conference on Artificial Intelligence and Machine Vision, AIMV, IEEE, 2025, pp. 1–4.

[92] M. Sadeghi, M. Kalantar, Decentralized parallel autonomous restoration: zone agents' automated rules, Int. J. Electr. Power Energy Syst. 171 (2025) 110992.

[93] A. Karimi, A.R. Abbasi, A fault-tolerant distributed agent-based framework for active/reactive power optimization using sensitivity gradient methods in smart grids, IEEE Access (2025).

[94] Y. Xiong, M. Huang, X. Liang, M. Tao, Resiliomate: A resilient multi-agent task executing framework for enhancing small language models, IEEE Access (2025).

[95] Y. Jin, J. Cai, J. Xu, Y. Huan, Y. Yan, B. Huang, Y. Guo, L. Zheng, Z. Zou, Self-aware distributed deep learning framework for heterogeneous iot edge devices, Future Gener. Comput. Syst. 125 (2021) 908–920.

[96] M.A. Naqvi, M. Astekin, S. Malik, L. Moonen, Adaptive immunity for software: towards autonomous self-healing systems, in: 2021 IEEE International Conference on Software Analysis Evolution and Reengineering, SANER, IEEE, 2021, pp. 521–525.

[97] M. Kim, Y. Kim, E. Lee, Denchmark: A bug benchmark of deep learning-related software, in: 2021 IEEE/ACM 18th International Conference on Mining Software Repositories, MSR, IEEE, 2021, pp. 540–544.

[98] M. Puvvadi, S.K. Arava, A. Santoria, S.S.P. Chennupati, H.V. Puvvadi, Coding agents: A comprehensive survey of automated bug fixing systems and benchmarks, in: 2025 IEEE 14th International Conference on Communication Systems and Network Technologies, CSNT, IEEE, 2025, pp. 680–686.

[99] F. Ribeiro, R. Abreu, J. Saraiva, Framing program repair as code completion, in: Proceedings of the Third International Workshop on Automated Program Repair, 2022, pp. 38–45.

[100] Z. Chen, S. Kommrusch, M. Monperrus, Neural transfer learning for repairing security vulnerabilities in c code, IEEE Trans. Softw. Eng. 49 (1) (2022) 147–165.

[101] B. Tang, B. Li, L. Bo, X. Wu, S. Cao, X. Sun, Grasp: Graph-to-sequence learning for automated program repair, in: 2021 IEEE 21st International Conference on Software Quality Reliability and Security, QRS, IEEE, 2021, pp. 819–828.

[102] Y. Ma, Q. Yang, R. Cao, B. Li, F. Huang, Y. Li, Alibaba lingmaagent: Improving automated issue resolution via comprehensive repository exploration, in: Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering, 2025, pp. 238–249.

[103] J. Yu, Y. Wu, Y. Zhan, W. Guo, Z. Xu, R. Lee, Co-learning: code learning for multi-agent reinforcement collaborative framework with conversational natural language interfaces, Front. Artif. Intell. 8 (2025) 1431003.

[104] P. Khlaisamniang, P. Khomduean, K. Saetan, S. Wonglapsuwan, Generative ai for self-healing systems, in: 2023 18th International Joint Symposium on Artificial Intelligence and Natural Language Processing, iSAI-NLP, IEEE, 2023, pp. 1–6.

[105] S. Kang, B. Chen, S. Yoo, J.-G. Lou, Explainable automated debugging via large language model-driven scientific debugging, Empir. Softw. Eng. 30 (2) (2025) 45.

[106] M. Jin, S. Shahriar, M. Tufano, X. Shi, S. Lu, N. Sundaresan, A. Svyatkovskiy, Inferfix: End-to-end program repair with llms, in: Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2023, pp. 1646–1656.

[107] B.-Y. Wu, U. Sharma, A. Rovinski, V.A. Chhabria, Openroad agent: An intelligent self-correcting script generator for openroad, in: 2025 IEEE International Conference on LLM-Aided Design, ICLAD, IEEE, 2025, pp. 16–22.

[108] J. Sohn, S. Kang, S. Yoo, Arachne: Search-based repair of deep neural networks, ACM Trans. Softw. Eng. Methodol. 32 (4) (2023) 1–26.

[109] A. Fernández-Miguel, S. Ortíz-Marcos, M. Jiménez-Calzado, A.P. Fernández del Hoyo, F.E. García-Muiña, D. Settembre-Blundo, Agentic ai in smart manufacturing: Enabling human-centric predictive maintenance ecosystems, Appl. Sci. 15 (21) (2025) 11414.

[110] P. Rondon, R. Wei, J. Cambronero, J. Cito, A. Sun, S. Sanyam, M. Tufano, S. Chandra, Evaluating agent-based program repair at google, 2025, arXiv preprint arXiv:2501.07531.

[111] Y. Wang, Z. Lin, Revisiting round-trip translation with llms and agentic translation, in: 2025 IEEE 5th International Conference on Software Engineering and Artificial Intelligence, SEAI, IEEE, 2025, pp. 172–178.

[112] I. Lorenzo-Fonseca, F. Maciá Pérez, A. Maciá-Fiteni, L. Arnau-Muñoz, Ai-driven multiagent iot system for energy consumption anomaly detection, IEEE Internet Things J. (2025) http://dx.doi.org/10.1109/JIOT.2025.3629154, 1–1.
[113] M.L.H. Souza, C.A. da Costa, G. de Oliveira Ramos, A machine-learning based data-oriented pipeline for prognosis and health management systems, Comput. Ind. 148 (2023) 103903.
[114] M. Esposito, A. Bakhtin, N. Ahmad, M. Robredo, R. Su, V. Lenarduzzi, D. Taibi, Autonomic microservice management via agentic ai and mape-k integration, in: European Conference on Software Architecture, Springer, 2025, pp. 105–118.
[115] T. Ahmed, P. Devanbu, Better patching using llm prompting, via self-consistency, in: 2023 38th IEEE/ACM International Conference on Automated Software Engineering, ASE, IEEE, 2023, pp. 1742–1746.
[116] S.-M. Chuang, C.-S. Chen, E.H. Wu, Implementation of non-intrusive intelligent sensor system and 5 g edge computing gateway for smart factory, in: 2022 IEEE 4th Eurasia Conference on IOT Communication and Engineering, ECICE, IEEE, 2022, pp. 64–69.
[117] M.-Y. Wan, Y. Xu, Z.-G. Wu, Attack-resilient distributed control of multi-agent systems under output feedback-driven multi-channel dos attacks, IEEE Trans. Autom. Sci. Eng. (2025).
[118] T.-C. Ureche, E.-C. Popovici, S. Halunga, L. Boicescu, D. Turcanu, A.P. Avasiloaie, Architecting secure smart infrastructures with ai microservices and autonomous agents: A state-of-the-art review and healthcare use case, in: 2025 IEEE International Black Sea Conference on Communications and Networking, BlackSeaCom, IEEE, 2025, pp. 1–6.
[119] I. Hamdeen, E.A. Badran, M.F. Kotb, M. Elgamal, Self-healing multi-agent techniques in electric power distribution systems: A review, Renew. Sustain. Energy Rev. 224 (2025) 116132.
[120] L. Yuan, Enhancing flexibility and reliability in smart distribution networks: A self-healing approach, Eksploat. I NiezawodnośĆ 27 (2) (2025).
[121] J. Feng, T. Yu, K. Zhang, L. Cheng, Integration of multi-agent systems and artificial intelligence in self-healing subway power supply systems: Advancements in fault diagnosis, isolation, and recovery, Processes 13 (4) (2025) 1144.
[122] O. Johnphill, A.S. Sadiq, F. Al-Obeidat, H. Al-Khateeb, M.A. Taheir, O. Kaiwartya, M. Ali, Self-healing in cyber–physical systems using machine learning: A critical analysis of theories and tools, Futur. Internet 15 (7) (2023) 244.
[123] H. Liang, X. Yin, Self-healing control: Review, framework, and prospect, IEEE Access 11 (2023) 79495–79512.
[124] P. Souza, T. Ferreto, R. Calheiros, Maintenance operations on cloud, edge, and iot environments: taxonomy, survey, and research challenges, ACM Comput. Surv. 56 (10) (2024) 1–38.
[125] O. Adeniyi, A.S. Sadiq, P. Pillai, M.A. Taheir, O. Kaiwartya, Proactive self-healing approaches in mobile edge computing: A systematic literature review, Computers 12 (3) (2023) 63.
[126] T.M. Tawfeeg, A. Yousif, A. Hassan, S.M. Alqhtani, R. Hamza, M.B. Bashir, A. Ali, Cloud dynamic load balancing and reactive fault tolerance techniques: a systematic literature review (slr), IEEE Access 10 (2022) 71853–71873.
[127] A. Ampatzoglou, S. Bibi, P. Avgeriou, A. Chatzigeorgiou, Guidelines for managing threats to validity of secondary studies in software engineering, in: Contemporary Empirical Methods in Software Engineering, Springer, 2020, pp. 415–441.
[128] H.N. Ho, E. Lee, Model-based reinforcement learning approach for planning in self-adaptive software system, in: Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication, 2015, pp. 1–8.