survey on surveying

Yuren Hao

December 2025

Abstract

This report proposes a taxonomy of literature collection methods used in recent systematic literature reviews (SLRs) in computer science and artificial intelligence. It organizes search and selection strategies into six main categories: (I) keyword-based multi-database search, (II) citation-based expansion, (III) scope restrictions, (IV) grey literature and web search, (V) tooling and advanced database features, and (VI) expert knowledge and manual additions. Within each category, it distinguishes subtypes such as Boolean block queries, synonym expansion, wildcards, and iterative refinement; backward/forward snowballing and citation-network analysis; filters on time window, language, document type, venue, and domain; structured use of web search and targeted grey sources; specialized tools for search, refinement, de-duplication, and screening; and expert-driven inclusion, relaxation of filters, and boundary judgements. Across categories, the report highlights typical trade-offs between recall, precision, scalability, transparency, and bias. In fast-evolving CS/AI domains, over-broad queries, overly strict scope restrictions, or uncritical reliance on a single index, tool, or expert community can each introduce systematic blind spots. The taxonomy is intended both as a descriptive map of current practice and as a checklist to help future SLRs combine complementary methods while explicitly documenting their choices and limitations.

1 Taxonomy of Literature Collection Methods in CS/AI SLRs

This section summarizes common literature collection methods used in recent systematic literature reviews (SLRs) in computer science and artificial intelligence. The taxonomy is organized in a three–level hierarchy: category $\rightarrow$ subcategory $\rightarrow$ concrete practices and survey examples.

1.1 Category I: Keyword-Based Multi-Database Search

1.1.1 Subcategory I.1: Boolean Combinations

Concrete practice. Search strings are decomposed into several "concept blocks". Within each block, synonyms are connected with logical `OR`, while different blocks are combined with logical `AND`. A typical query has a block for the core technology, another block for the application domain, and another block for the task or perspective. The same logical structure is executed across multiple digital libraries (e.g., IEEE Xplore, ACM Digital Library, Scopus, Web of Science, SpringerLink).

Survey examples.

- **Deep learning in software engineering.** Watson et al. design a generic block-level query such as `("Deep" OR "Learning" OR "Neural")` and run it in IEEE Xplore, ACM Digital Library, SpringerLink and DBLP, using advanced-search filters to restrict by year and venue.[15] Their unified search string across four digital libraries initially returned 1,699 records; after applying venue filters, screening, and snowballing, they ended up with 128 primary studies.[15]
- **Blockchain-based applications.** Casino et al. search Scopus using the term "blockchain" in the title field, then iteratively refine the result set using built-in filters (subject area, document type, etc.), effectively implementing a structured Boolean filtering pipeline.[4]
- **AI in higher education.** Zawacki-Richter et al. use a multi-block Boolean query of the form `("artificial intelligence" OR "AI" OR "machine learning") AND ("higher education" OR "university" ...)` across large databases to retrieve studies on AI in higher education.[17]

1.1.2 Subcategory I.2: Synonym and Phrase Expansion

Concrete practice. For each conceptual block, authors explicitly enumerate synonyms, acronyms, and related phrases to increase recall. All surface forms for one concept (for example, "artificial intelligence", "AI", "machine learning", "generative AI") are grouped and joined with `OR`. Different concept groups (technology, domain, type of study, and so on) are then combined with `AND` to target the intersection of relevant facets.

Survey examples.

- **Blockchain adoption.** AlShamsi et al. use a broad adoption-related block such as `("adoption" OR "acceptance" OR "use" OR "intention to use" OR "continued use")` in conjunction with the core term "blockchain" across several libraries (IEEE Xplore, ACM DL, Scopus, Springer, Emerald, MDPI, Google Scholar), explicitly treating all these phrases as alternative realizations of the same adoption concept.[1]
- **AI in higher education / education.** Zawacki-Richter et al. and other AI-in-education SLRs expand the AI block to include phrases such as "artificial intelligence", "AI", and "machine learning" and combine these with an education block ("education", "learning", "teaching", and so on), as well as with blocks indicating empirical work ("empirical study", "case study", and so forth).[17]
- **Fake news detection.** Thompson et al. construct a synonym-rich query for fake news, combining phrases such as "fake news detection", "false news", and `("fake news" AND "social media")` or `("fake news" AND ("internet" OR "online"))` to cover diverse terminologies around fake news on online platforms.[13]

1.1.3 Subcategory I.3: Wildcards and Truncation

Concrete practice. Wildcards (for example, the asterisk `*`) are used to capture morphological variants of a term, such as:

- `microserv*` to match `microservice`, `microservices`, `micro-service`;
- `applicat*` to match `application`, `applications`;
- `secur*` to match `security`, `secure`, `securing`.

This reduces the need to enumerate each inflected form explicitly, while still preserving the block structure of the Boolean query.

Survey examples.

- **Microservice security.** Berardi et al. work with patterns such as `microserv*` and `secur*` when searching IEEE Xplore, ACM DL, and Scopus, ensuring that all common variants of "microservice" and "security" are captured.[2] Their systematic search yielded 1,067 studies, from which only 46 were ultimately retained as primary studies after full-text screening and quality assessment.[2]
- **Explainable AI applications.** In many XAI application reviews, authors use truncation in the application-related block, for example `application*` or `process*`, to include both singular and plural forms in a single compact query. A typical example is the use of `("explainable artificial intelligence" OR XAI) AND (application* OR process*)` in Web of Science.

1.1.4 Subcategory I.4: Adaptive and Iterative Query Refinement

Concrete practice. Searches are often conducted in multiple waves. An initial query is executed, yielding a first candidate set. After manually inspecting part of this set, reviewers may identify additional terminology or missing concepts and refine the query, running follow-up searches. In some cases, the same query is re-run months or years later to update an earlier SLR, or to capture papers published after the first search.

Survey examples.

- **Blockchain-based applications.** Casino et al. run an initial Scopus search and then perform an updated search a few months later using the same core term "blockchain" together with refined filters, in order to capture newly published studies and ensure that the review reflects the very recent explosion of blockchain research.[4]
- **Machine learning for predictive maintenance.** Carvalho et al. focus on modern machine learning methods for predictive maintenance and implicitly adopt an iterative perspective by confining the analysis to studies where such methods are actually applied. Pilot searches and inspection of early results inform the final set of ML methods and predictive-maintenance terms used to construct the search strategy.[3]

Remarks. Overall, keyword-based multi-database search is the backbone of most SLRs in CS/AI: it is relatively easy to reproduce and scales well across large digital libraries. However, it is also sensitive to vocabulary mismatch and to the choice of synonyms and truncations. In practice, over-broad queries can produce thousands of hits (for example, the 1,699 initial records in Watson et al. before narrowing to 128 primary studies), while over-narrow queries can silently miss entire streams of work.[15] In rapidly evolving CS/AI domains with unstable terminology, poorly tuned queries and inconsistent use of wildcards or synonyms are especially common sources of bias and missed coverage.

1.2 Category II: Citation-Based Expansion (Snowballing and Networks)

1.2.1 Subcategory II.1: Pure Backward Snowballing

Concrete practice. Backward snowballing starts from the set of studies included after the initial database search and systematically scans their reference lists. Every cited paper is considered as a candidate and is screened using the same inclusion/exclusion criteria. This process can be repeated for newly added studies to capture older foundational work that is not directly retrieved by the keyword search.

Survey examples.

- **Deep learning in software engineering.** Watson et al. perform backward snowballing on all studies that passed the initial selection. They inspect every reference of each included paper and add any referenced work that also meets their inclusion criteria, thereby capturing additional relevant studies that were missed by keyword searches.[15] Starting from 1,699 initial hits, this combination of venue filters, screening, and snowballing yields 128 primary studies.[15]
- **Microservice security.** Berardi et al. apply backward snowballing in multiple rounds, starting from the initially selected microservice-security papers and gradually expanding the set by recursively inspecting reference lists. Their overall search (database search plus snowballing) yields 1,067 records, of which only 46 ultimately qualify as primary studies.[2]

1.2.2 Subcategory II.2: Combined Backward and Forward Snowballing

Concrete practice. Some SLRs use both backward and forward citation chasing. In addition to inspecting references, reviewers look up papers that cite the included studies, using tools such as Google Scholar, Scopus, or Web of Science. Both classes of candidates are screened against the same inclusion criteria, and the expansion may be repeated in several iterations.

Survey examples.

- **Analysing app reviews for software engineering.** Dabrowski et al. augment their database-based corpus of app-review studies with a snowballing procedure. They first perform backward snowballing over the reference lists of all included papers and then carry out a targeted forward snowballing step on highly cited papers, adding further relevant studies on app reviews for software engineering.[5] Through this systematic procedure (database search plus snowballing), they ultimately include 182 papers published between 2012 and 2020.[5]
- **Defect prediction with deep learning.** In their review of deep learning for software defect prediction, Giray et al. start from database searches and then use forward snowballing to extend the set. They report a final pool of 102 peer-reviewed studies, selected and analyzed after combining database search with citation-based expansion.[7]

1.2.3 Subcategory II.3: Citation Network and Bibliometric Analysis

Concrete practice. Rather than manually chasing individual references, some SLRs construct an explicit citation network, where nodes are papers and edges represent citation links. They then apply network analysis techniques such as main-path analysis, co-citation analysis, bibliographic coupling, or community detection to identify clusters and seminal works. These clusters provide a data-driven way to structure the literature and to ensure that influential papers in each cluster are included.

Survey examples.

- **Circular economy literature.** Khitous et al. use citation network analysis and main-path analysis to uncover existing themes and emerging trends in circular-economy research, tracing the development of the literature over time and revealing the seminal papers along the main citation paths.[9]
- **AI in innovation research.** Mariani et al. complement a traditional SLR with several bibliometric techniques (citation analysis, co-citation analysis, bibliographic coupling, co-word analysis) to map the intellectual and conceptual structure of AI in innovation research. This helps identify clusters of studies and influential works within each cluster.[10]
- **AI in marketing.** Verma et al. conduct a systematic review of artificial intelligence in marketing and use citation-based and co-occurrence analyses to organize the field into thematic streams, ensuring that central and frequently co-cited works are represented in their synthesis.[14]

Remarks. Citation-based expansion is powerful for discovering influential work that does not match the exact search string (for example, older but seminal papers, or studies with unusual terminology). In CS/AI, where new terms and acronyms appear rapidly, snowballing and citation-network analysis are often critical to avoid missing key contributions. The main risks are that snowballing can cause an uncontrolled growth of candidate papers and can bias the corpus toward already well-cited sub-communities. Reviews that start from very broad queries (such as the 1,699 initial hits of Watson et al. or the 1,067 hits of Berardi et al.)[15, 2] must carefully manage this growth, or else screening costs and topic drift can become prohibitive.

1.3 Category III: Scope Restrictions

1.3.1 Subcategory III.1: Time Window Filters

Concrete practice. SLRs often restrict the publication year of included studies to a specific range, for example to the last decade or to a fixed interval such as 2008–2019. This focuses the review on recent developments or on a well-defined era, and keeps the size of the corpus manageable.

Survey examples.

- **Blockchain-based applications.** Casino et al. explicitly emphasize covering blockchain applications published in the "last decade" in high-ranked scientific journals, effectively restricting their time window to the roughly ten years prior to the review.[4]
- **Machine learning for predictive maintenance.** Carvalho et al. focus on modern machine learning methods for predictive maintenance, implicitly confining their corpus to studies from recent decades where such ML techniques are applied.[3]
- **Fake news detection approaches.** De Beer and Matthee restrict inclusion to studies published between 2008 and 2019, reflecting the emergence of social-media platforms and online fake news in that period.[6]
- **IoT in healthcare.** Jawad et al. set an explicit time window from 2015 to 2022 for IoT-enabled healthcare studies, capturing the surge of IoT research in medical contexts.[8]

1.3.2 Subcategory III.2: Language and Document-Type Restrictions

Concrete practice. It is common to limit the corpus to a single language (almost always English) and to specific document types (for example, peer-reviewed journal articles, or journal plus conference papers), while excluding theses, non-peer-reviewed material, and non-English texts. Dedicated grey-literature SLRs define their own allowed document types.

Survey examples.

- **Blockchain in healthcare.** Tandon et al. review blockchain in healthcare using only English-language journal articles, excluding other publication types.[12]
- **Fake news detection approaches.** De Beer and Matthee include only studies that are written in English, published in IT or technology journals or conference proceedings, and that have at least a given citation count, thereby enforcing both language and quality-related document-type filters.[6]
- **Microservices grey-literature review.** Soldani et al. focus exclusively on specific grey-literature document types—blog posts, white papers, industrial magazines, and videos—and combine inclusion/exclusion criteria with quality assessment tailored to these formats.[11]

1.3.3 Subcategory III.3: Venue and Source Targeting

Concrete practice. Reviewers often restrict the set of venues from which they accept papers. For example, they may include only top-tier conferences and journals in a particular discipline, or only specific domain journals, or conversely only grey-literature outlets.

Survey examples.

- **Deep learning in software engineering.** Watson et al. predefine a list of top software engineering and programming languages conferences and journals, as well as selected machine learning and AI venues. Only papers published in these venues are considered, enforcing a strict venue-quality filter.[15]
- **Blockchain-based applications.** Casino et al. include research published in high-ranked scientific journals combined with a small number of authoritative reports, thereby targeting a specific tier of venues.[4]
- **Fake news detection approaches.** De Beer and Matthee include only IT and technology journals or conferences, deliberately excluding non-technical outlets in media studies or political science to keep the focus on computational approaches.[6]
- **Microservices grey literature.** Soldani et al. reverse the usual pattern by limiting the review to industry sources (blogs, white papers, magazines, videos), not academic venues, in order to capture practitioners' perspectives on microservices in industry.[11]

1.3.4 Subcategory III.4: Domain and Topic Scoping

Concrete practice. Inclusion and exclusion criteria are used to narrow the topical focus to a specific subdomain. Papers that are only tangentially related to the topic are excluded. For example, an SLR on fake news detection may exclude papers that discuss fake news from a political or sociological perspective but do not propose computational methods for detection.

Survey examples.

- **Fake news detection.** De Beer and Matthee require that included papers have their main focus on fake news on digital platforms and on the identification (detection) of fake news. Studies on fake news impacts or non-technical discussions are excluded.[6]
- **Blockchain in healthcare.** Tandon et al. and Yaqoob et al. consider only works where blockchain is applied in healthcare settings (for example, medical records, data sharing), and exclude purely financial or cryptocurrency-focused studies.[12, 16]
- **Microservices in industry.** Soldani et al. include only sources that (i) discuss the industrial application of microservices and (ii) explicitly address the benefits or shortcomings of microservice design, development, or operation, thereby tightly scoping the domain to "pains and gains" of adopting microservices.[11]

Remarks. Scope restrictions are essential for keeping SLRs tractable and focused. Time-window filters, venue targeting, and domain scoping help prevent the situation where thousands of marginally relevant papers must be screened. However, in CS/AI they can easily hide important work: early but influential papers can fall outside a nominal time range, and cross-disciplinary contributions (for example, fake news work in social-science venues, or healthcare-blockchain papers in medical journals) may be missed by strict venue constraints.[6, 12] Good practice is to combine explicit filters with later citation-based and expert checks to rescue genuinely canonical studies that lie just outside the initial scope.

1.4 Category IV: Grey Literature and Web Search

1.4.1 Subcategory IV.1: Systematic Web Search with Top-K Results

Concrete practice. Reviewers use general-purpose search engines (for example Google or Bing) with carefully chosen query strings and then systematically inspect only the top K results. Each result is screened against inclusion and exclusion criteria, typically to identify industry reports, policy documents, and other non-indexed sources.

Survey examples.

- **Blockchain-based applications.** Casino et al. conduct web searches using combinations of "blockchain" and "application", systematically checking the first 200 hits returned by Google. From these, they identify relevant government or institutional reports and other grey literature describing specific blockchain applications.[4]

1.4.2 Subcategory IV.2: Targeted Grey-Literature Sources

Concrete practice. Instead of searching the entire web, some SLRs define a fixed set of grey sources (for example specific technology blogs, company engineering blogs, trade magazines, or video platforms) and conduct keyword search within these sources.

Survey examples.

- **Microservices in industry.** Soldani et al. limit their search to grey-literature outlets such as technical blogs, white papers, industrial magazines, and videos (for example InfoQ, company tech blogs, conference talks on video platforms), where practitioners discuss microservices experiences in detail.[11]

1.4.3 Subcategory IV.3: Separate Analysis of Academic vs. Grey Literature

Concrete practice. Some reviews explicitly treat academic literature and grey literature as disjoint subsets, with separate coding and analysis pipelines for each subset. The two are later compared to identify gaps between research and practice.

Survey examples.

- **Blockchain-based applications.** Casino et al. distinguish between high-ranked journal articles and grey literature (for example reports). Grey sources are required to describe specific blockchain applications, rather than only background information, and are analyzed alongside but distinctly from academic studies.[4]
- **Microservices grey-literature review.** Soldani et al. perform a systematic review entirely on grey literature and compare their findings with the academic microservices literature, yielding insights into where practice reports problems and benefits that have not yet been fully explored in research.[11]

Remarks. Grey literature and web search are crucial in CS/AI for capturing industrial practice, emerging technologies, and vendor-driven trends that have not yet reached academic venues. The downside is weaker quality control and poorer reproducibility: web search is sensitive to personalization and time, and blogs or white papers can disappear. Pitfalls include quietly over-weighting a few charismatic blog posts, or mixing unvetted engineering claims with peer-reviewed evidence without clearly separating the two strata.

1.5 Category V: Tooling and Advanced Database Features

1.5.1 Subcategory V.1: Dedicated Search Tools (e.g., Publish or Perish)

Concrete practice. Tools such as Publish or Perish are used to programmatically query Google Scholar and export result sets for further filtering. The exported results can then be filtered by adding domain-specific keywords or by applying other criteria such as venue or year.

Survey examples.

- **Deep learning in software engineering.** Watson et al. use Publish or Perish to query Google Scholar and then apply software-engineering keywords to the returned results, complementing the searches performed in the four major digital libraries.[15]

1.5.2 Subcategory V.2: Advanced Database Features (Refinement and Related Documents)

Concrete practice. Reviewers leverage advanced functionalities provided by digital libraries, such as subject-area filters, document-type filters, "related documents" or "similar articles" recommendations, and field restrictions (for example, searching only in titles, abstracts, or keywords). Starting from a broad result set, they iteratively refine it to a more focused corpus.

Survey examples.

- **Blockchain-based applications.** Casino et al. extensively use Scopus refinement features (subject area, document type, and related-documents search) to progressively narrow down the initial set of blockchain-related records to a focused and deduplicated corpus.[4]
- **Microservice security.** Berardi et al. use advanced search features in IEEE Xplore, ACM DL, and Scopus to restrict results by fields and to target studies that combine microservices and security concerns.[2]

1.5.3 Subcategory V.3: Reference Management and Screening Platforms

Concrete practice. Reference managers (for example EndNote, Zotero, Mendeley) are used for de-duplication, and screening platforms (for example Rayyan, Covidence) are used for collaborative screening. Many SLRs report their process via a PRISMA flow diagram, tracking the number of records at each stage: identified, screened, eligible, and finally included.

Survey examples.

- **Microservice security.** Berardi et al. employ PRISMA-style tracking and reference-management tools to handle the large number of microservice-security records gathered from multiple databases.[2]
- **App-review analysis.** Dabrowski et al. rely on reference-management and screening tools to de-duplicate and screen the corpus of app-review studies in a transparent, reproducible way.[5]

Remarks. Tool support improves scalability and transparency: automated capture from Google Scholar or Scopus, de-duplication, collaborative screening, and PRISMA diagrams all make it easier to defend the review process. The main pitfall in CS/AI is over-reliance on a single tool or index (for example, only Google Scholar via Publish or Perish), which can bias the corpus and hide coverage gaps in venues not well indexed. Poor documentation of tool versions, search-settings, and export dates also makes it hard for others to reproduce or update the review later.

1.6 Category VI: Expert Knowledge and Manual Additions

1.6.1 Subcategory VI.1: Manual Inclusion of Canonical Papers

Concrete practice. After automated search and snowballing, domain experts may identify a small set of canonical papers that are considered indispensable for the topic, even if they fall outside some of the formal scope restrictions such as time window or venue. These papers are then manually added to the final corpus.

Survey examples.

- **Deep learning in software engineering.** Watson et al. manually add a key arXiv paper on deep learning for software engineering based on their domain knowledge and evidence of its influence, even though it does not strictly satisfy all venue or time-window constraints, thus ensuring that a seminal but borderline paper is not omitted.[15]

1.6.2 Subcategory VI.2: Relaxing Filters for Seminal Works

Concrete practice. Scope filters such as time window and venue can be treated as soft constraints for highly influential works. During snowballing or citation analysis, if a study is found to be seminal (for example through high citation counts or repeated reference by many included papers), the reviewers may relax filters to include it, even when it lies slightly outside the nominal constraints.

Survey examples.

- **Fake news and AI-in-education SLRs.** In practice, SLR authors frequently relax their time-window restrictions for a small number of canonical papers (for example early but highly cited works on fake-news detection or foundational AI-in-education studies), even when these lie slightly outside the nominal year range, so that the historical context of the field is properly represented.[6, 17]

1.6.3 Subcategory VI.3: Expert Boundary Judgement for Edge Cases

Concrete practice. Some candidate studies sit at the boundary of the topic (for example generic blockchain papers with minor mention of healthcare, or generic microservices papers with a brief discussion of operations). For such edge cases, inclusion and exclusion decisions are often made through expert judgement, based on whether the study substantially contributes to the research questions.

Survey examples.

- **Blockchain in healthcare and microservices in industry.** In SLRs on blockchain in healthcare and on microservices in industry, reviewers use their domain knowledge to decide whether borderline studies (for example papers that only peripherally mention healthcare or only superficially mention microservice "pains" and "gains") should be included, beyond what simple keyword matching can reliably determine.[12, 16, 11]

Remarks. Expert knowledge is invaluable for plugging gaps left by mechanical search and for ensuring that genuinely foundational work is represented. At the same time, it is the least transparent and most subjective component of the pipeline. In CS/AI, where many authors are themselves active contributors to the field they survey, there is a real risk of confirmation bias: experts may over-include their own work or that of their immediate community, and under-include competing lines of research. Clear documentation of when and why expert overrides are applied, and keeping the number of manual additions small compared to the total (for example, one or two canonical papers out of over a hundred primary studies), helps keep this step defensible.

References

[1] Mohammed AlShamsi, Mostafa Al-Emran, and Khaled Shaalan. A systematic review on blockchain adoption. *Applied Sciences*, 12(9):4245, 2022.

[2] Davide Berardi, Saverio Giallorenzo, Jacopo Mauro, Andrea Melis, Fabrizio Montesi, and Marco Prandini. Microservice security: A systematic literature review. *PeerJ Computer Science*, 8:e779, 2022.

[3] Thyago P. Carvalho, Fabrízzio A. A. M. N. Soares, Roberto Vita, Roberto da P. Francisco, João P. Basto, and Symone G. S. Alcalá. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137:106024, 2019.

[4] Fran Casino, Thomas K. Dasaklis, and Constantinos Patsakis. A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Telematics and Informatics*, 36:55–81, 2019.

[5] Jacek Dąbrowski, Emmanuel Letier, Anna Perini, and Angelo Susi. Analysing app reviews for software engineering: A systematic literature review. *Empirical Software Engineering*, 27(2):43, 2022.

[6] Dylan de Beer and Machdel Matthee. Approaches to identify fake news: A systematic literature review. In *Integrated Science in Digital Age 2020*, volume 136 of *Lecture Notes in Networks and Systems*, pages 13–22. Springer, 2020.

[7] Görkem Giray, Kwabena Ebo Bennin, Ömer Köksal, Önder Babur, and Bedir Tekinerdogan. On the use of deep learning in software defect prediction. *Journal of Systems and Software*, 195:111537, 2023.

[8] Huda Hussein Mohamad Jawad, Zainuddin Bin Hassan, Bilal Bahaa Zaidan, Farah Hussein Mohammed Jawad, Duha Husein Mohamed Jawad, and Wajdi Hamza Dawod Alredany. A systematic literature review of enabling IoT in healthcare: Motivations, challenges, and recommendations. *Electronics*, 11(19):3223, 2022.

[9] Fatima Khitous, Fernanda Strozzi, Andrea Urbinati, and Fernando Alberti. A systematic literature network analysis of existing themes and emerging research trends in circular economy. *Sustainability*, 12(4):1633, 2020.

[10] Marcello M. Mariani, Isa Machado, Vittoria Magrelli, and Yogesh K. Dwivedi. Artificial intelligence in innovation research: A systematic review, conceptual framework, and future research directions. *Technovation*, 122:102623, 2023.

[11] Jacopo Soldani, Damian Andrew Tamburri, and Willem-Jan van den Heuvel. The pains and gains of microservices: A systematic grey literature review. *Journal of Systems and Software*, 146:215–232, 2018.

[12] Anushree Tandon, Amandeep Dhir, A. K. M. Najmul Islam, and Matti Mäntymäki. Blockchain in healthcare: A systematic literature review, synthesizing framework and future research agenda. *Computers in Industry*, 122:103290, 2020.

[13] Robyn C. Thompson, Seena Joseph, and Timothy T. Adeliyi. A systematic literature review and meta-analysis of studies on online fake news detection. *Information*, 13(11):527, 2022.

[14] Sanjeev Verma, Rohit Sharma, Subhamay Deb, and Debojit Maitra. Artificial intelligence in marketing: Systematic review and future research direction. *International Journal of Information Management Data Insights*, 1:100002, 2021.

[15] Cody Watson, Nathan Cooper, David Nader Palacio, Kevin Moran, and Denys Poshyvanyk. A systematic literature review on the use of deep learning in software engineering research. *ACM Transactions on Software Engineering and Methodology*, 31(2):1–58, 2022.

[16] Sobia Yaqoob, Muhammad Murad Khan, Ramzan Talib, Arslan Dawood Butt, Sohaib Saleem, Fatima Arif, and Amna Nadeem. Use of blockchain in healthcare: A systematic literature review. *International Journal of Advanced Computer Science and Applications*, 10(5):644–653, 2019.

[17] Olaf Zawacki-Richter, Victoria I. Marín, Melissa Bond, and Franziska Gouverneur. Systematic review of research on artificial intelligence applications in higher education – where are the educators? *International Journal of Educational Technology in Higher Education*, 16(1):39, 2019.