Journal of Internet Services and Applications, 2026, 17:1, doi: 10.5753/jisa.2026.6589

A Systematic Mapping Study of Container-Based Virtualization Technologies

José A. Arias-Pinzón [**Universidad del Quindío** | ***josea.ariasp@uqvirtual.edu.co***]
Anobis H. Correa-Urbano [**Universidad del Quindío** | ***anobish.correau@uqvirtual.edu.co***]
Luis E. Sepúlveda-Rodríguez [**Universidad del Quindío** | ***lesepulveda@uniquindio.edu.co***]
Christian A. Candela-Uribe [**Universidad del Quindío**| ***christiancandela@uniquindio.edu.co***]

Facultad de ingeniería, Universidad del Quindío, Carrera 15 #12N, Armenia, Quindío, 630001, Colombia.

Received: 20 September 2025 • **Accepted:** 12 May 2026 • **Published:** 25 June 2026

Abstract Container-based virtualization (CBV) is widely adopted in modern IT infrastructure, with technologies such as Docker and Kubernetes commonly used for application packaging and orchestration, respectively. Despite the growing body of research in this domain, no existing secondary study systematically maps CBV technologies across both IT domains and academic dimensions—comprising education, research, and outreach. This paper presents a Systematic Mapping Study (SMS) of 226 primary studies published between 2022 and 2024, identified through a hybrid strategy combining database searches across five digital libraries and forward/backward snowballing. Studies were classified using 11 IT domains and three academic dimensions, assessed through three quality indices (CVI, SCI, IRRQ), and organized into a dual-axis taxonomic structure. The results reveal a marked concentration around Docker (41.6%) and Kubernetes (29.6%), a dominance of IT Infrastructure as a research domain (75.66%), and an underrepresentation of education (11.06%) and outreach (5.88%) in the CBV literature. Six concrete research gaps are identified, including the need for alternative runtime evaluation, orchestration beyond Kubernetes, and empirical studies on CBV in educational contexts. The proposed taxonomy and identified gaps provide a structured foundation for researchers, educators, and practitioners navigating the evolving CBV landscape.

Keywords: Container-based virtualization, Cloud computing, Docker, IT infrastructure, Kubernetes, Software engineering, Systematic mapping study.

1 Introduction

Cloud computing has become a widely adopted paradigm in contemporary information technology, enabling scalable and resilient solutions through on-demand resource provisioning (Segun-Falade *et al.*, 2024). Among the foundational technologies supporting this paradigm, container-based virtualization (CBV) has gained wide adoption due to its lightweight footprint, portability, and rapid deployment capabilities (Almoudane, 2025). Unlike full virtualization, which emulates complete hardware environments, containers share the host operating system kernel while maintaining application isolation, resulting in substantially lower overhead (Kozhirbayev and Sinnott, 2017). These properties have positioned CBV as a commonly used approach for deploying, managing, and scaling applications across distributed environments, facilitating continuous integration, agile development, and microservices architectures (Clement, 2025).

Within the CBV ecosystem, Docker has emerged as the *de facto* standard for container creation and management, while Kubernetes dominates container orchestration. However, the rapid evolution of this field has produced a diverse landscape of alternative technologies—including Podman, Singularity, LXC, gVisor, and Kata Containers—each optimized for specific use cases such as rootless operation, high-performance computing, or enhanced security (Baresi *et al.*, 2024a). This technological diversification necessitates a systematic analysis to identify which technologies are adopted in which contexts and to what extent.

Beyond industrial applications, CBV technologies have increasingly permeated academic settings, supporting education through reproducible laboratory environments, enabling research through portable computational pipelines, and facilitating outreach through accessible cloud-based platforms. However, the extent and nature of this academic adoption remain poorly characterized in the literature.

This paper presents a Systematic Mapping Study (SMS) that addresses this gap by mapping 226 primary studies (2022–2024) across 11 IT domains and three academic dimensions (education, research, outreach). The study contributes: *(1)* a comprehensive, reproducible mapping of the CBV research landscape; *(2)* a proposed dual-axis taxonomic structure linking technologies to both IT domains and academic impact; and *(3)* the identification of six concrete research gaps with actionable future directions.

The remainder of this paper is organized as follows: Section 2 outlines the study motivation. Section 3 analyzes related works. Section 4 describes the SMS methodology. Section 5 addresses threats to validity. Section 6 presents the analysis, discussion, and future research directions. Section 7 concludes the paper.

2 Motivation

The adoption of cloud computing has produced a wide range of solutions based on container-based virtualization Hassan *et al*. (2022). However, as noted by multiple authors Waseem *et al*. (2024); Vhatkar and Bhole (2022); Kithulwatta *et al*. (2022b), the literature remains fragmented, particularly in terms of thematic scope and research focus: the high volume of publications makes it difficult to identify clear usage patterns, benefits, and limitations across application domains.

This study is motivated by three specific needs. First, there is no existing secondary study that simultaneously maps CBV technologies across multiple IT domains *and* academic dimensions (education, research, outreach). Second, the rapid proliferation of container technologies beyond Docker—including Podman, Singularity, gVisor, and others—requires a systematic assessment of their adoption and research coverage. Third, the potential of CBV as a transversal tool for academic activities (reproducible research, portable teaching environments, accessible outreach platforms) has not been systematically evaluated.

The expected outcomes include: *(i)* a comprehensive map of CBV research trends across 11 IT domains; *(ii)* a classification of studies by their contribution to education, research, and outreach; and *(iii)* a taxonomic structure that can guide technological decision-making for researchers, educators, and practitioners.

3 Related Works

Several secondary studies have addressed container-based virtualization (CBV) from different perspectives. To position the contribution of the present SMS, this section analyzes the existing literature organized by three dimensions: *(i)* scope and coverage, *(ii)* methodology, and *(iii)* domain focus. Table 1 synthesizes this comparative analysis.

Prior reviews differ substantially in the breadth of technologies and application contexts examined. Bentaleb *et al*. (2022a) and Sepúlveda-Rodríguez *et al*. (2022) both propose taxonomic classifications of virtualization technologies; however, neither extends its analysis to the academic dimension (education, research, and outreach). Similarly, Malhotra *et al*. (2024a) provide a systematic literature review focused exclusively on container lifecycle management—image detection, scheduling, security, and performance—without mapping these concerns to specific IT domains or educational contexts. In contrast, Kaiser *et al*. (2022a, 2023b) narrow their scope to ARM-compatible container technologies, prioritizing energy efficiency and edge computing performance. While valuable, this architectural focus limits the generalizability of their findings across the full spectrum of CBV use cases.

Despite these contributions, the existing literature remains methodologically fragmented in its treatment of CBV. Naydenov and Ruseva (2023) adopt a systematic mapping methodology focusing on container orchestration architectures in cloud computing; however, their categorization scheme is restricted to orchestration and does not encompass runtime technologies, academic applications, or cross-domain analysis. The remaining studies employ narrative or traditional review methods, which—while informative—lack the structured, reproducible search protocols and quality assessment mechanisms that characterize an SMS Kitchenham *et al*. (2010).

A common limitation across all reviewed studies is the absence of a cross-cutting analysis that maps CBV technologies to both IT domains (e.g., software development, HPC, security, AI) and academic dimensions simultaneously. None of the existing works: *(a)* provides a comprehensive mapping of CBV across multiple IT domains; *(b)* examines the role of containerization in education and outreach; or *(c)* offers a taxonomic structure linking technologies, domains, and academic impact.

Table 1. Comparative analysis of related secondary studies. Multi-domain: covers multiple IT domains; Acad.: includes academic dimensions; Taxon.: proposes a taxonomy; Reprod.: provides reproducibility artifacts.

Study	Type	Multi-domain	Acad.	Taxon.	Reprod.
Bentaleb *et al*. (2022a)	Review	✗	✗	✓	✗
Kaiser *et al*. (2022a)	Review	✗	✗	✗	✗
Sepúlveda-Rodríguez *et al*. (2022)	Review	✗	✗	✓	✗
Kaiser *et al*. (2023b)	Review	✗	✗	✗	✗
Naydenov and Ruseva (2023)	SMS	✗	✗	✓	Partial
Malhotra *et al*. (2024a)	SLR	✗	✗	✗	Partial
This study	**SMS**	✓	✓	✓	✓

This SMS addresses these gaps by providing: *(1)* a systematic, reproducible mapping of 226 primary studies across 11 IT domains; *(2)* a systematic classification linking CBV technologies to education, research, and outreach; and *(3)* a taxonomic structure that integrates technological and academic perspectives, enabling researchers and practitioners to identify both consolidated areas and under-explored research opportunities.

4 Review Method

This study follows a Systematic Mapping Study (SMS) methodology, guided by the established frameworks of Petersen et al. Runeson and Höst (2009) and Kitchenham and Charters Kitchenham *et al*. (2010). Unlike a Systematic Literature Review (SLR), which aims to synthesize evidence on a specific question, an SMS provides a broad overview of a research area by classifying and categorizing existing literature Runeson and Höst (2009). Following Mourao *et al*. (2017) and Nguyen *et al*. (2015), a hybrid search approach was adopted, combining automated database queries with manual snowballing to maximize coverage.

To ensure transparency and reproducibility, the SMS-Builder tool Candela-Uribe *et al*. (2022) was employed throughout the process for study identification, classification, data extraction, and quality assessment. The SMS process comprises six stages: (1) planning, (2) study search, (3) quality assessment, (4) data extraction, (5) study classification, and (6) results. Figure 1 illustrates these stages.

4.1 Planning

The planning stage established the research goals, questions, metrics, classification criteria, and quality assessment indices. Figure 2 summarizes the components of this stage.

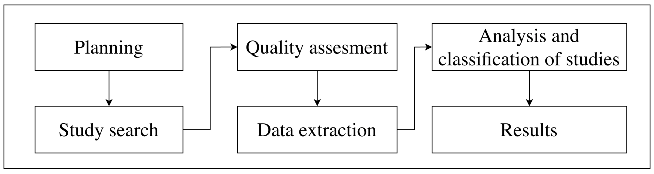

Figure 1. Stages of the SMS process

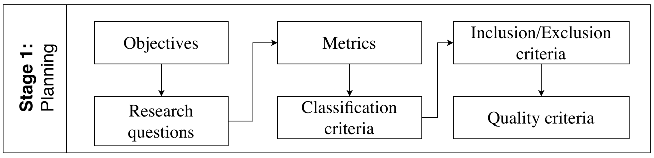

Figure 2. Composition of the planning stage

4.1.1 Study Goals

Two overarching goals were defined to guide this SMS, as presented in Table 2.

Table 2. Goals of the study

Goal	Description
G1	Identify studies related to CBV in education, research, and outreach.
G2	Classify studies related to CBV across IT domains, including software development, computational thinking, parallel computing, data analysis, artificial intelligence, computer networks, IT infrastructure, HPC, security, cloud computing, and blockchain.

4.1.2 Research Questions

The research questions were formulated using the GQM (*Goal Question Metric*) framework Needleman (2002) and the PICOC model Petticrew and Roberts (2008), which structures the population, intervention, comparison, outcome, and context of the study (Table 4). Two research questions were derived, as detailed in Table 3.

4.1.3 Metrics

Quantitative metrics were defined to measure the distribution of studies across the classification structure (Table 6). The search period was restricted to 2022–2024 to ensure currency.

Table 4. PICOC model specification

Aspect	Description
Population	Studies related to CBV applied across IT domains, with emphasis on education, research, and outreach.
Intervention	Identification and classification of CBV studies within established IT domains.
Comparison	1. Comparison of CBV projects by reported success rates across IT domains. 2. Analysis of CBV impact on academic activities relative to alternative solutions.
Outcome	Classification structure mapping CBV studies to IT domains and academic dimensions.
Context	Education, research, and outreach contexts adopting CBV technologies.

4.1.4 Research Topics

Based on the research questions and PICOC model, four research topics were defined: *Container-based virtualization*, *Education*, *Research*, and *Industry*. These topics were further refined through the IT domains identified as relevant to the study scope.

4.1.5 Inclusion and Exclusion Criteria

Table 5 presents the inclusion and exclusion criteria. The three-year window (2022–2024) balances currency with sufficient volume. Both journal articles and conference proceedings were included to capture the full publication landscape in this rapidly evolving field.

Table 6. Metrics defined for the analysis

Metric	Description
M1	Number of studies identified per IT domain.
M2	Number of studies classified under education.
M3	Number of studies classified under research.
M4	Number of studies classified under outreach.

4.1.6 Quality Assessment Criteria

Three quality indices were defined to evaluate the relevance and rigor of the selected studies.

Content Validity Index (CVI). The CVI assesses the degree to which each study aligns with the SMS objectives, adapted from established content validity methodology Almanasreh *et al.* (2019); YAGHMAEI (2003). Each study was independently rated by K evaluators (where K is odd, to prevent ties) on a scale from 0 (no relevance) to 5 (high relevance). Following the proportion-based CVI approach Almanasreh *et al.* (2019), we define the item-level CVI (I-CVI) as the proportion of evaluators who rate a study above a relevance threshold t:

$$\text{I-CVI} = \frac{n_t}{K} \tag{1}$$

where n_t is the number of evaluators assigning a score $\geq t$ (with $t = 3$ adopted in this study), and K is the total number of evaluators. An I-CVI ≥ 0.78 indicates acceptable content validity Almanasreh *et al.* (2019). For aggregation across the study corpus, the Scale-level CVI based on the average method (S-CVI/Ave) is computed as the mean of all I-CVI values.

Scientific Citation Index (SCI). The SCI captures citation-normalized impact relative to publication recency. For a study with C citations accumulated between 2022 and 2024, published A years before the extraction date:

$$\text{SCI} = \frac{C}{A} \tag{2}$$

This normalization ensures that recently published studies with emerging citation counts are not penalized relative to older, more-cited works.

Index of Relationship to Research Questions (IRRQ). The IRRQ quantifies the coverage of a study with respect to the defined research questions. Given $Q = 2$ research questions in this SMS, the IRRQ for a study addressing n questions is:

$$\text{IRRQ} = \frac{n}{Q} \tag{3}$$

where $n \in \{0, 1, 2\}$ and $Q = 2$. Thus, IRRQ $\in \{0, 0.5, 1\}$, where 1 indicates full coverage of both research questions. This index enables identification of studies with broad versus narrow thematic relevance.

Table 3. Research questions and their motivation

Goal	Question	Description	Motivation
G1	Q1	Which studies related to container-based virtualization (CBV) technologies contribute to education, research, and outreach?	CBV enables environment reproducibility, facilitating the transfer of IT solutions across contexts. Understanding its academic penetration can inform cross-domain research.
G2	Q2	Which primary studies related to CBV technologies contribute to IT domains such as software development, HPC, AI, security, cloud computing, and others?	The goal is to provide a structured overview of CBV adoption across IT domains, enabling researchers and practitioners to identify trends without requiring exhaustive primary analysis.

Table 5. Inclusion and exclusion criteria

Category	Inclusion	Exclusion
Screening field	Abstract	—
Publication type	Journal articles and conference proceedings	Theses, book chapters, grey literature
Discipline	Computer Science, Information Technology, Engineering, IT Management	Disciplines unrelated to virtualization or computing
Time period	2022–2024	Before 2022
Language	English	Non-English publications

4.2 Stage 2: Study Search

A hybrid search strategy combining database queries and snowballing was employed. Figure 3 summarizes the components of this stage.

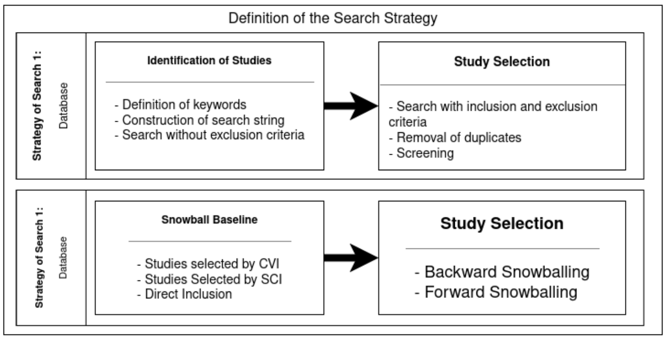

Figure 3. Composition of the study search stage

4.2.1 Defining the Search Strategy

Two complementary strategies were combined. The first involves automated search string execution in academic databases Jalali and Wohlin (2012). The second, snowballing, identifies additional studies through backward (reference tracking) and forward (citation tracking) analysis of a seed set Jalali and Wohlin (2012); Goodman (1961).

4.2.2 Search Strategy 1: Databases

This strategy comprises two phases: *Study Identification* (search string construction and execution) and *Study Selection* (criteria-based refinement).

- **Study Identification:** Five databases were queried: *ACM*, *IEEE Xplore*, *Springer*, *Science Direct*, and *Taylor & Francis*. Keywords were derived from the PICOC model (Table 7) and expanded with synonyms (Table 8). Boolean operators (*AND*, *OR*) and exact-phrase matching were used to construct database-specific search strings. Complete search strings are available via the reproducibility artifacts (Section 4.6.2).
  Execution across all databases yielded **6,530** preliminary results, with Springer contributing the largest share (**4,562**; 69.8%). Table 9 details the distribution.

Table 7. Keywords identified using the PICOC model

Aspect	Description
Population	CBV, IT Domains, Education, Research, Outreach
Intervention	Identification, Classification
Comparison	Success rate, Evidence of use
Output	Classification of CBV studies per IT domain
Context	Education, Research, Outreach

Table 8. Keywords for database search

Keyword	Synonyms
Container-based virtualization	Application virtualization, Docker, Lightweight Virtualization
Education	Education System, Education Development, Higher Education
Research	Research Group, Research Proposal
Industry	IT Services, Technology Infrastructure, Cloud Computing

- **Study Selection:** Applying inclusion and exclusion criteria reduced the set to **976** studies (Table 10). After removing **274** duplicates, title, abstract, and keyword screening excluded **593** irrelevant studies, yielding **110** selected studies. Figure 4 summarizes this process.

4.2.3 Search Strategy 2: Snowballing

The snowballing strategy comprised two phases: *Baseline Construction*, which established a seed set from the database results using the CVI, SCI, and direct inclusion criteria; and *Study Selection*, which extended coverage through backward (reference) and forward (citation) analysis of each seed article Wohlin (2014). Baseline construction started from the **110** database results. From this set, **25** articles were selected using the SCI criterion (first quartile by citation count), which provides an objective, evaluator-independent measure of academic impact. Direct inclusion was additionally applied as a complementary strategy to capture relevant works not indexed or not retrievable through the automated queries Jalali and Wohlin (2012). All candidate studies were evaluated against the same inclusion and exclusion criteria used in the standard protocol; of the initial set of known studies considered, only one satisfied all requirements and was retained, bringing the baseline total to **26** studies. Figure 5 summarizes this process. Forward snowballing via Google Scholar Ali *et al.* (2019) identified **495** new articles; backward snowballing yielded **87**. After removing **14** duplicates and applying the same title, abstract, and keyword screening, **116** studies were selected through this strategy. Figure 6 summarizes the process.

Table 9. Search results per database using keywords

Criterion	ACM	IEEE	Science Direct	Springer	Taylor and Francis
Search results using keywords only	189	426	4562	353	1000
Contribution percentage	2.89%	6.52%	69.86%	5.4%	15.31%

Table 10. Search results per database using keywords after applying inclusion/exclusion criteria

Criterion	ACM	IEEE	Science Direct	Springer	Taylor and Francis
Search results after applying keywords only	48	134	46	592	156
Contribution percentage	4.91%	13.72%	4.71%	60.65%	15.98%

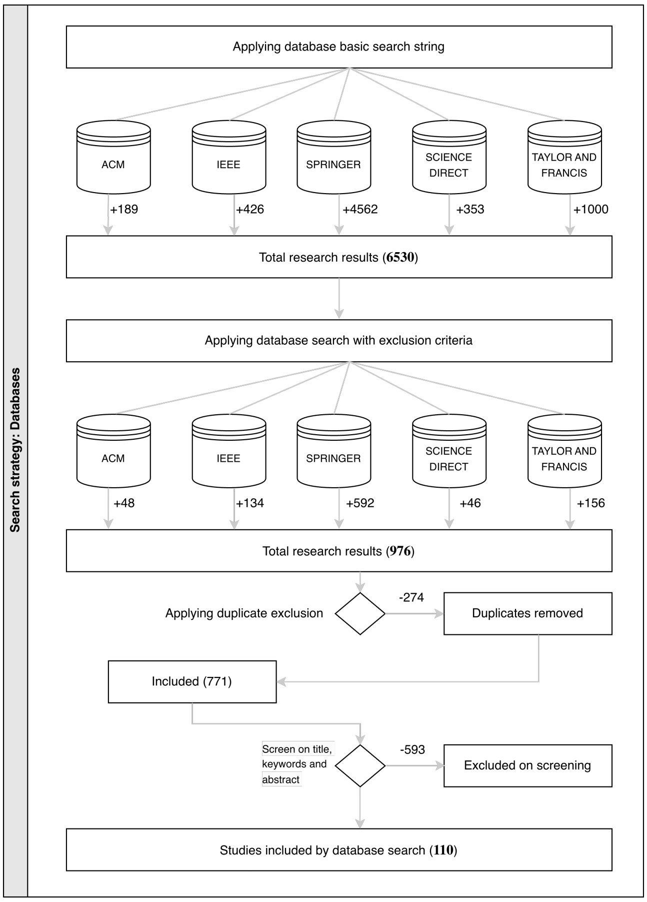

Figure 4. Summary of activities and results obtained in the database search strategy

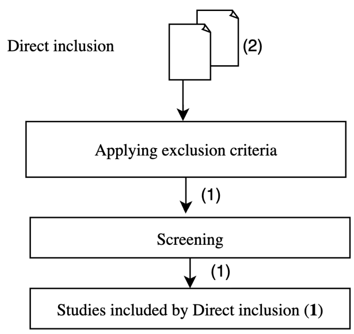

Figure 5. Summary of the direct inclusion search strategy

4.2.4 Results of the Study Search

The combined search yielded **226** primary studies: **110** from databases, **115** from snowballing, and **1** through direct inclusion. The nearly-equal split between strategies (Table 11) confirms the complementarity of the hybrid approach.

Table 11. Results of the study search

Strategy	Studies	%
Databases	110	48.67%
Snowballing	115	50.88%
Direct Inclusion	1	0.44%
Total	**226**	**100%**

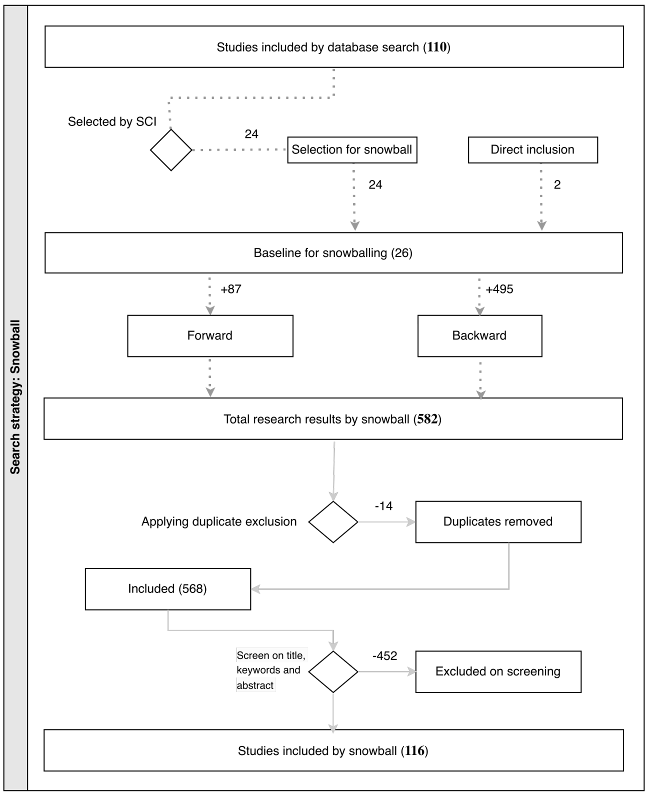

Figure 6. Summary of the snowballing search strategy

4.3 Stage 3: Quality Assessment

Although quality assessment is not mandatory in an SMS Ali *et al.* (2019), incorporating it strengthens the rigor of the mapping and brings it closer to a systematic review Wohlin (2014). Three complementary indices—CVI, SCI, and IRRQ—were applied to evaluate study relevance.

4.3.1 Content Validity Assessment (CVI)

Each study was independently rated by an odd number of evaluators ($K \geq 3$) on a 0–5 relevance scale. The proportion-based I-CVI (Equation 1) was computed for each study, and studies with I-CVI ≥ 0.78 were considered to have acceptable content validity. Two assessment rounds were conducted: the first during baseline construction for snowballing (Section 4.2.4), and the second after all 226 studies were identified, with results reported in Section Study Classification.

4.3.2 Citation-Based Quality Assessment (SCI)

The SCI (Equation 2) was computed using citation data from Google Scholar and the SMS-Builder tool Candela-Uribe *et al*. (2020). A frequency analysis identified the top quartile (Q1) of studies by SCI, representing those with the highest citation-normalized impact.

4.3.3 Research Question Coverage Assessment (IRRQ)

The IRRQ (Equation 3) was computed for each study based on its thematic alignment with Q1 and Q2, as determined through the classification process. Studies with IRRQ $= 1$ (addressing both research questions) were identified through frequency analysis as the most thematically comprehensive.

4.4 Stage 4: Data Extraction

After completing study search and quality assessment, **226** primary studies were identified and labeled SPS001 through SPS226. The complete list is provided in Table 12 (Appendix A).

4.5 Stage 5: Study Classification

The SPS were classified using the topics defined during planning (Table 13, provided in Appendix B). A single SPS may be associated with multiple topics; for example, SPS069 is classified under IT Infrastructure, Security, Cloud Computing, and Research. This multi-label approach reflects the interdisciplinary nature of CBV research and enables cross-domain analysis.

Following classification, each SPS was evaluated using the CVI, SCI, and IRRQ quality indices. Tables 14, 15, and 16 (Appendix C) present the top-quartile studies for each index, disaggregated by topic and year.

4.6 Stage 6: Results

This section presents and interprets the findings from the SMS, organized in three parts: (1) an overview of the SPS corpus with source and temporal analysis, (2) a technology and domain distribution analysis with interpretation of observed trends, and (3) a keyword co-occurrence analysis.

4.6.1 Overview of the SPS Corpus

The SMS identified **226** selected primary studies (SPS), listed in Table 12 (Appendix A). The following subsections analyze the distribution of these studies across sources, technologies, academic dimensions, and quality indices.

Source and strategy distribution. Table 17 (Appendix D) presents the classification of CBV technologies by academic dimension. Docker is the most frequently represented technology across education, research, and outreach, accounting for **100** SPS (44.24%). This predominance reflects Docker's mature ecosystem, extensive documentation, and low barrier to entry, which collectively facilitate adoption in academic contexts where ease of deployment is prioritized over specialized performance characteristics.

Table 18 maps IT domains to academic dimensions (provided in Appendix D for completeness). IT Infrastructure is the most represented domain, with **171** SPS (75.66%), indicating that containerization research remains strongly anchored to infrastructure-level concerns such as deployment, scaling, and resource management.

Technology landscape analysis. Figure 7 shows the distribution of studies by source and strategy. Of the database-sourced studies (110 SPS), IEEE Xplore and ACM Digital Library jointly contribute 68.18%, reflecting the strong alignment of CBV research with computing-focused venues. The snowballing strategy (115 SPS) was dominated by forward snowballing (92.17%), suggesting that CBV is an expanding field where newer publications actively cite foundational works.

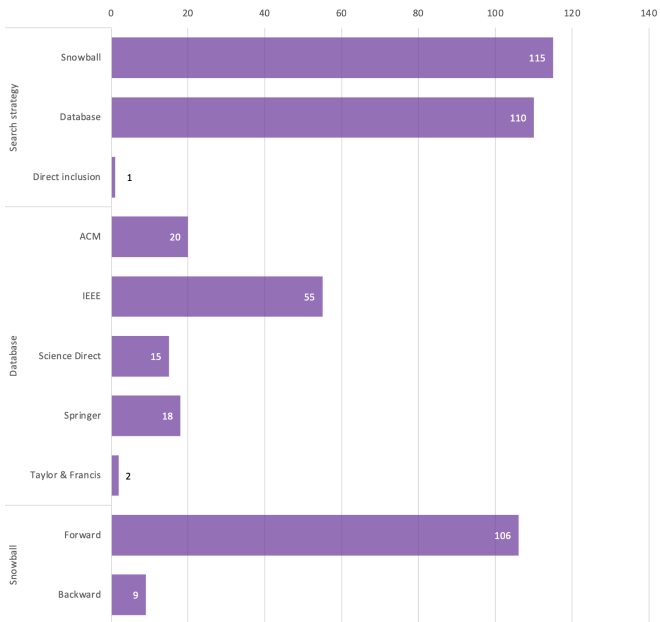

Figure 7. SPS by source and search strategy

Figure 8 reveals the distribution of container runtime technologies. Docker leads with **94** SPS, followed distantly by Podman (7), LXC and Containerd (4 each), and Singularity, runC, and gVisor (3 each). This concentration indicates that *despite the growing ecosystem of alternative container runtimes designed for security (gVisor, Kata Containers), HPC (Singularity), and rootless operation (Podman), the research community remains heavily Docker-centric*. This gap between technological diversity and research coverage represents an opportunity for future studies to evaluate emerging runtimes in domains where Docker's limitations are well documented.

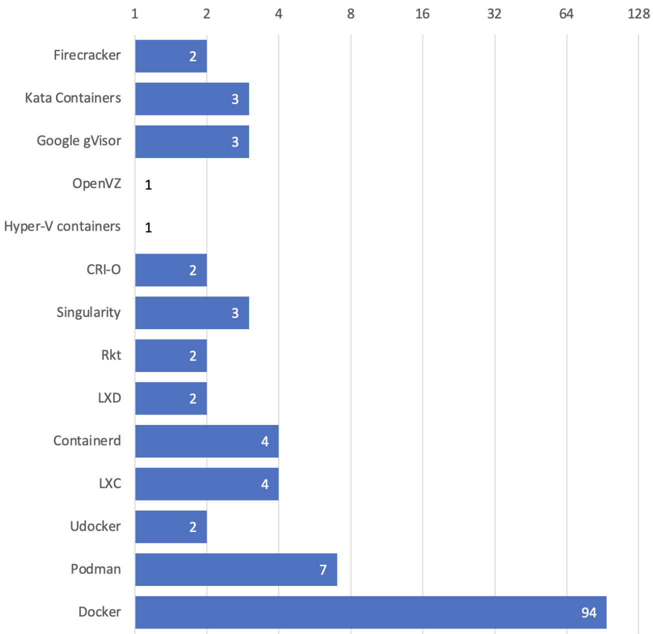

Figure 8. Distribution of container runtime technologies across SPS

Figure 9 shows orchestrator distribution. Kubernetes leads with **67** SPS, consistent with its status as the *de facto* standard for container orchestration. Docker Swarm (9 SPS) and Apache Mesos (5 SPS) trail significantly. The marginal representation of alternatives such as OpenShift (2), Docker Compose (3), and cloud-native services (Amazon ECS/EKS, 1 each) suggests that *academic research has not yet systematically evaluated the trade-offs between Kubernetes and its alternatives*, particularly in edge computing, serverless, and resource-constrained environments where lighter orchestration solutions may be more appropriate.

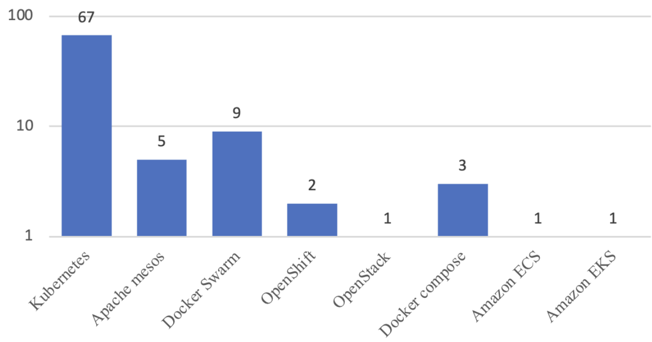

Figure 9. Distribution of orchestrator technologies across SPS

Academic dimension analysis. Figure 10 illustrates the intersection of studies across academic dimensions. Research dominates with **187** exclusive SPS, while Education (19 exclusive) and Outreach (8 exclusive) remain underrepresented. Notably, *no identified study simultaneously addresses all three academic dimensions*, indicating a fragmentation in academic production. Only 6 SPS bridge Research and Education, and 6 bridge Research and Outreach, with zero overlap between Education and Outreach. This finding suggests that the potential of CBV as a transversal tool linking teaching, research output, and societal impact remains largely unexplored.

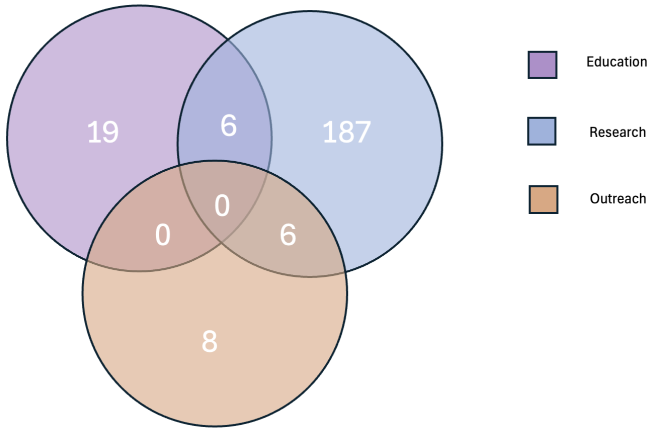

Figure 10. Venn diagram of SPS across academic dimensions

Topic and temporal distribution. Figure 11 shows topic distribution per research question. For Q1, Research accounts for 83.61% of studies, while Outreach represents only 5.88%—a disparity that reflects the limited adoption of containerization in community engagement and societal applications. For Q2, IT Infrastructure leads (41.28%), followed by Cloud Computing (14.37%), while Blockchain (0.76%) and Parallel Computing represent emerging but underexplored intersections with CBV.

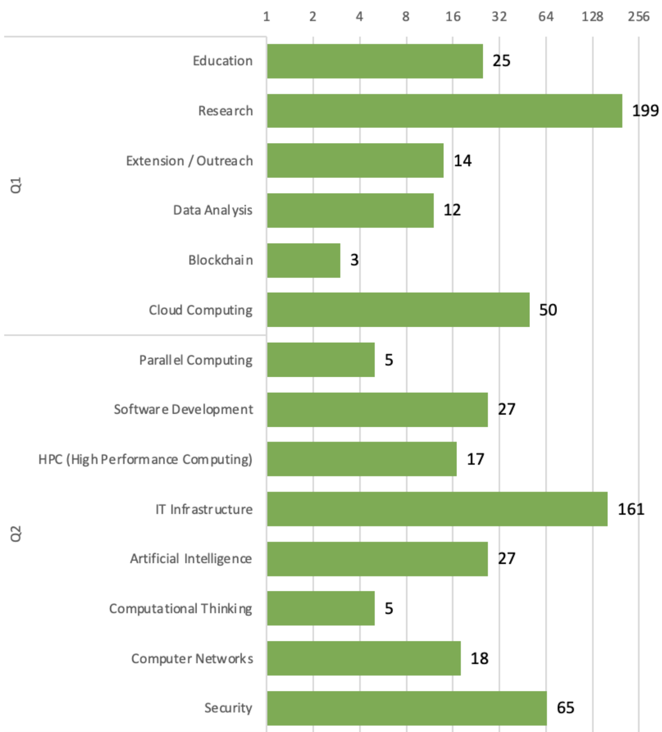

Figure 11. SPS distribution by research questions and topics

The temporal analysis (Figure 12) reveals sustained growth, from **49** SPS in 2022 to **107** in 2024 (a 118% cumulative increase). The sharpest growth occurred between 2023 and 2024 (+52.85%), coinciding with the broader adoption of Kubernetes-based cloud-native architectures and the proliferation of edge computing applications. The CVI index shows an upward trend (from 7 to 9, +28.57%), suggesting that more recent studies exhibit stronger alignment with the SMS objectives. The SCI index remains stable around 18, with a slight recovery in 2024 (+11.76% over 2023), while the IRRQ index exhibits sustained growth from 30 to 39 (+30% cumulative), indicating increasing thematic breadth in newer publications.

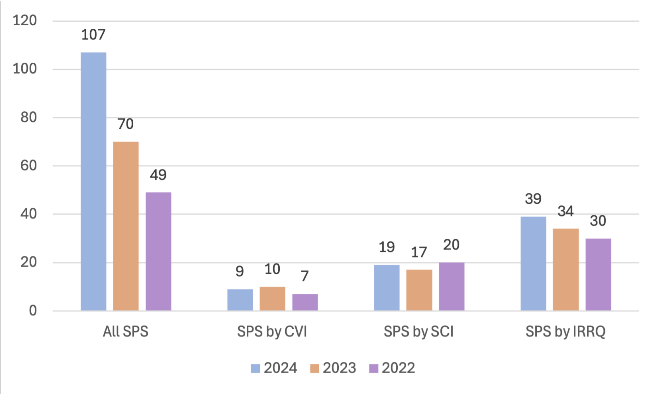

Figure 12. SPS by year and quality indices

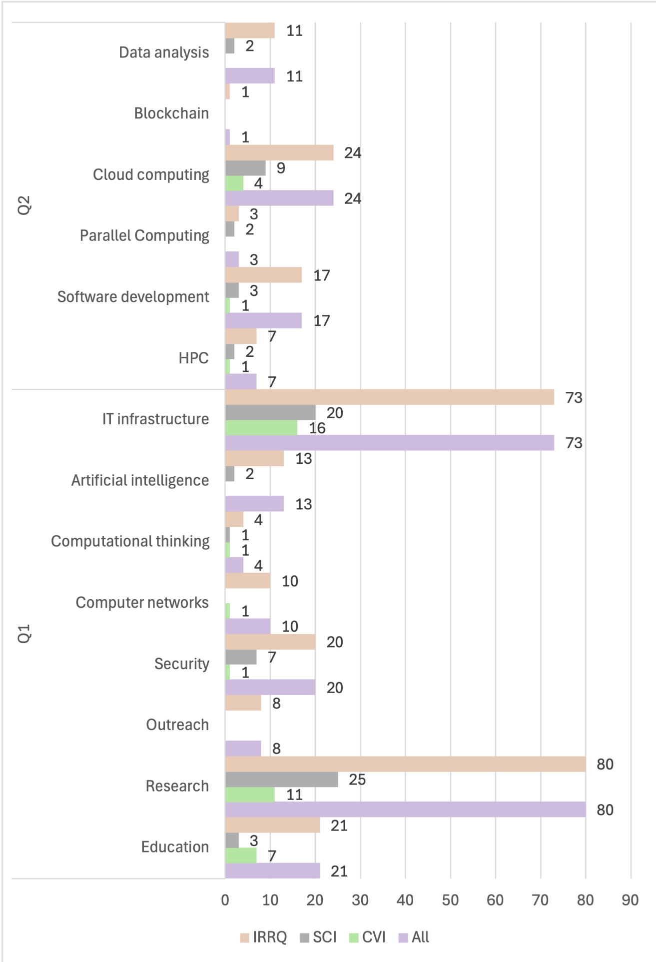

Figure 13. SPS by quality indices, topics, and research questions

Figure 13 presents the quality indices disaggregated by topic. IT Infrastructure not only has the highest volume (73 SPS, 43.71% of Q2) but also concentrates the highest-quality studies across all three indices, reflecting its centrality in the CBV research landscape. The Outreach topic, with only 8 SPS (7.33% of Q1), represents the least represented topic in this mapping.

Figure 14 presents a cross-analysis of keywords. The term ‘‘*Container*’’ enables the identification of 57 SPS, while education-related keywords (*Learning*, *Cybersecurity education*) appear in only 2 SPS each—suggesting that a shared vocabulary linking CBV to educational and outreach applications has not yet emerged.

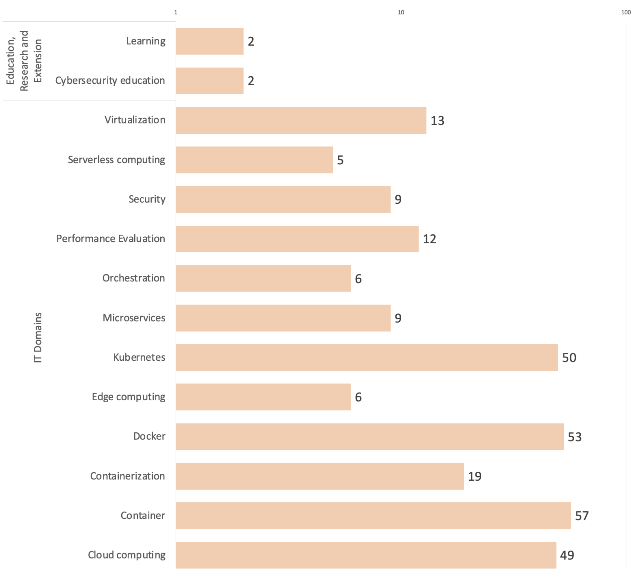

Figure 14. SPS keyword co-occurrence analysis

4.6.2 *Reproducibility*

To ensure full reproducibility, two verification mechanisms are provided:

1) A public SMS-Builder instance containing all process data: https://sms-vbc.iti.grid.uniquindio.edu.co/sms.xhtml. Credentials: ‘‘*invitado*’’ for both username and password.
2) A Docker image integrating all required documentation: https://hub.docker.com/r/anubis1001/tg-vbc-sms-builder.

5 Threats to Validity

Four categories of threats to validity are identified, along with the mitigation strategies employed.

5.1 Selection Bias

Seven measures were implemented to mitigate selection bias. First, the SMS followed established guidelines Runeson and Höst (2009); Kitchenham *et al*. (2010), including GQM and PICOC frameworks. Second, five major databases were queried. Third, synonyms were included for all key terms to ensure broad coverage. Fourth, search strings were iteratively refined through pilot searches. Fifth, a hybrid strategy combining database search with snowballing increased coverage. Sixth, alert systems (Endnote, Mendeley, Google Scholar) monitored for newly published studies. Seventh, three quality indices (CVI, SCI, IRRQ) provided complementary assessment perspectives. The CVI and IRRQ indices carry inherent subjectivity; this was mitigated through collaborative evaluation by an odd number of independent evaluators ($K \geq 3$).

5.2 Classification Errors

We classified studies according to the topics defined during planning, corresponding to CBV technologies, IT domains, education, research, and outreach. Multi-topic classification was permitted when a study's scope spanned multiple areas. All classifications underwent peer review by an odd number of evaluators to reduce individual bias.

5.3 Data Extraction Inaccuracy

The SMS-Builder software Candela-Uribe *et al*. (2020) was used for structured data extraction, minimizing manual processing errors. Peer review was conducted on extracted data following the recommendations of Kitchenham and Charters Kitchenham *et al*. (2010).

5.4 Search Protocol Errors

The search protocol was executed under peer review: one evaluator implemented the protocol while a second independently verified the process. SMS-Builder was used throughout to reduce manual data handling and ensure process consistency.

6 Analysis, Discussion, and Future Research Directions

The SMS results reveal several cross-cutting patterns that merit interpretation beyond descriptive statistics. This section synthesizes the key findings, proposes a taxonomic structure, identifies research gaps, and outlines concrete future research directions.

6.1 Proposed Taxonomic Structure

The classification of 226 SPS across IT domains and academic dimensions motivates a taxonomic structure (Figure 15) that organizes the CBV research landscape along two axes: *(i)* the IT domain (software development, IT infrastructure, cloud computing, HPC, security, AI, etc.) and *(ii)* the academic dimension (education, research, outreach). This dual-axis taxonomy represents a contribution of this study, as no prior secondary study has simultaneously mapped both dimensions. The taxonomy can serve as a decision-support tool for researchers identifying relevant literature, for educators designing container-based curricula, and for practitioners selecting technologies aligned with domain-specific requirements.

6.2 Key Findings and Interpretation

The analysis of the 226 selected primary studies (SPS) reveals several cross-cutting patterns that require interpretation beyond descriptive statistics.

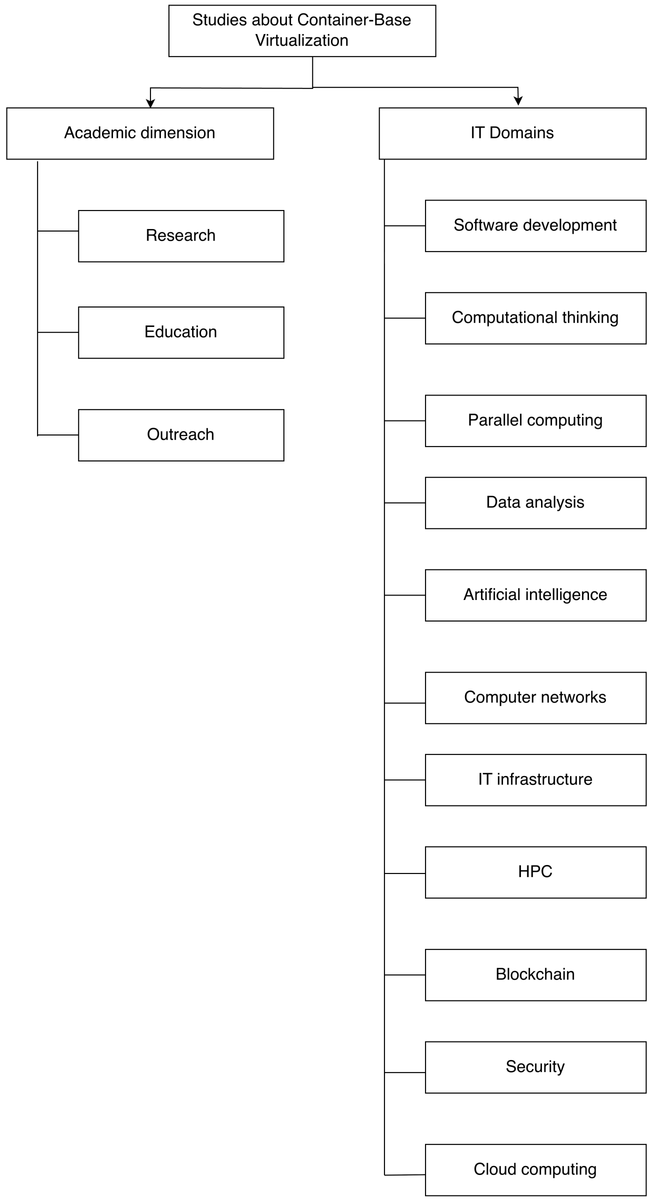

Figure 15. Proposed taxonomic structure for CBV research

Technology concentration and its implications. A primary observation is the pronounced concentration of Docker (94 SPS, 41.6%) and Kubernetes (67 SPS) within the current research corpus. While Docker's prevalence is attributable to its early market entry and ecosystem maturity, it implies that existing research conclusions may not be directly generalizable to alternative runtimes. Specifically, emerging technologies with distinct security models (e.g., gVisor, Kata Containers), rootless permission structures (e.g., Podman), or optimizations for High-Performance Computing (e.g., Singularity) remain underrepresented. Consequently, researchers must exercise caution when extrapolating findings from Docker-centric studies to specialized deployment contexts where these alternatives offer superior performance or isolation.

The IT Infrastructure bias. The mapping identifies a pronounced bias toward the IT Infrastructure domain, which accounts for 75.66% of the analyzed studies. This concentration underscores that containerization research is still heavily anchored to fundamental management concerns such as deployment, scaling, and resource allocation. In contrast, critical emerging domains—including Blockchain (0.76%), Parallel Computing (1.77%), and Computational Thinking (2.21%)—require further investigation despite their clear potential for container-based applications. This disparity suggests a missed opportunity for developing reproducible testing environments in blockchain or portable frameworks for parallel computing.

The academic dimension gap. A further finding is the fragmentation across academic spheres. While Research dominates the literature (83.61%), Outreach accounts for only 5.88% of the corpus. This imbalance suggests that the research community has yet to fully exploit the core properties of container-based virtualization (CBV)—namely portability and environment isolation—to facilitate technology transfer and societal impact. Furthermore, the total absence of studies simultaneously addressing Education, Research, and Outreach indicates that the transversal potential of CBV as a tool for integrated academic activities remains unrealized.

Temporal trends and maturation signals. The 118% growth in publications from 2022 to 2024, combined with the increasing CVI and IRRQ indices, indicates both growing interest and improving methodological alignment with the field's core questions. However, the stable SCI around 18 suggests that while more studies are published, citation impact has not proportionally increased—a pattern consistent with a rapidly expanding but potentially fragmenting research area.

6.3 Identified Research Gaps

Based on the SMS findings, the following research gaps are identified:

RG1: **Alternative runtime evaluation.** Systematic comparative studies of container runtimes beyond Docker (e.g., Podman, Singularity, gVisor, Kata Containers) across performance, security, and usability dimensions are needed.

RG2: **Orchestration beyond Kubernetes.** Research evaluating lightweight orchestration alternatives (K3s, Nomad, Docker Compose) for edge computing, IoT, and resource-constrained environments is underrepresented.

RG3: **CBV in education and outreach.** Empirical studies measuring the impact of containerization on learning outcomes, curriculum design, and outreach program effectiveness are nearly absent from the literature.

RG4: **Cross-domain integration.** No identified study bridges all three academic dimensions simultaneously, presenting an opportunity for holistic CBV adoption frameworks.

RG5: **Emerging IT domains.** The intersection of CBV with blockchain, quantum computing simulation, and computational thinking requires dedicated investigation.

RG6: **Security of container ecosystems.** While security appears in 65 SPS, most address container isolation rather than supply-chain security, image provenance, or runtime attestation—areas of growing practical concern.

6.4 Future Research Directions

Building on the identified gaps, the following concrete research directions are proposed:

- **Benchmarking frameworks:** Development of standardized benchmarking methodologies for comparing container runtimes and orchestrators across heterogeneous hardware (x86, ARM, RISC-V) and workload profiles (HPC, microservices, AI training).
- **Educational impact studies:** Controlled experiments measuring the effect of container-based laboratory environments on student learning outcomes, engagement, and skill transferability in computer science education.
- **Outreach and knowledge transfer:** Design and evaluation of container-packaged educational platforms that facilitate technology transfer from universities to industry and communities, particularly in resource-limited settings.
- **Security posture analysis:** Comprehensive studies on container supply-chain security, including image scanning effectiveness, Software Bill of Materials (SBOM) adoption, and runtime security monitoring in production environments.
- **Lightweight orchestration for edge/IoT:** Empirical evaluation of Kubernetes alternatives in edge and IoT deployments where resource constraints and latency requirements differ from cloud-native assumptions.

6.5 Implications for Research and Practice

For *researchers*, this SMS provides a structured entry point into the CBV literature, enabling targeted investigation of underexplored domains and informed positioning of new contributions. The identified gaps (RG1–RG6) offer concrete starting points for future studies. For *practitioners*, the technology distribution analysis highlights both the safety of Docker/Kubernetes adoption (given extensive research backing) and the risk of overlooking better-suited alternatives for specific use cases. For *educators*, the near-absence of containerization in formal educational frameworks represents an opportunity to develop pedagogical approaches leveraging CBV's reproducibility and portability.

7 Conclusions

This paper presented a Systematic Mapping Study (SMS) on container-based virtualization (CBV) technologies, analyzing 226 primary studies published between 2022 and 2024. By employing a hybrid search strategy and rigorous quality assessment indices (CVI, SCI, IRRQ), the study established a comprehensive overview of the current research landscape. The evidence synthesized leads to the following conclusions:

1. **Comprehensive mapping:** This SMS is, to the best of the authors' knowledge, the first secondary study to simultaneously map CBV technologies across both IT domains and academic dimensions (education, research, and outreach), addressing a gap identified in the reviewed literature.

2. **Technology monoculture:** A pronounced concentration around Docker (41.6%) and Kubernetes (29.6%) was observed. This concentration suggests that the research literature may not adequately represent the technical characteristics of alternative runtimes like Podman or Singularity, and lighter orchestrators such as K3s or Nomad.
3. **Domain imbalance:** Research remains heavily anchored to IT Infrastructure (75.66%), while critical areas such as Blockchain (0.76%) and Parallel Computing (1.77%) represent ''blind spots'' that require further investigation.
4. **Academic fragmentation:** Although CBV is inherently suited for environment reproducibility, its adoption is heavily skewed toward Research (83.61%), with Education (11.06%) and Outreach (5.88%) remaining underrepresented. The fact that no study addressed all three dimensions simultaneously indicates that CBV's transversal potential for holistic academic impact remains largely unrealized.
5. **Methodological contribution:** The proposed dual-axis taxonomy provides a structured reference for researchers and practitioners to navigate the rapidly expanding CBV field. Furthermore, the identification of six concrete research gaps (RG1–RG6) provides directions for future efforts in benchmarking, security, and pedagogical development.

The sustained growth in publication rates—a 118% increase over the three-year period—indicates continued interest in the field but also suggests a risk of literature fragmentation. The taxonomic structure and reproducibility artifacts provided herein aim to support future work by offering a structured foundation for the next generation of cloud-native research.

Authors' Contributions

All authors contributed equally to the conceptualization, methodology, investigation, formal analysis, writing — original draft, writing — review and editing, and visualization of this article. All authors read and approved the final version of the manuscript.

José A. Arias-Pinzón
Systems and Computing Engineer
Universidad del Quindío

Anobis H. Correa-Urbano
Systems and Computing Engineer
Universidad del Quindío

Luis E. Sepúlveda-Rodríguez
Systems and Computing Engineer
Ph.D. in Engineering
Universidad del Quindío

Christian A. Candela-Uribe
Systems and Computing Engineer
Ph.D. in Engineering
Universidad del Quindío

Competing Interests

The authors declare that they have no competing interests.

Availability of Data and Materials

The data and materials supporting the findings of this study are available upon reasonable request from the corresponding author. All resources for this study were provided by the authors and Universidad del Quindío.

References

Abas, S. U., Duran, F., and Tekerek, A. (2023). A raspberry pi based blockchain application on iot security. *Expert Systems with Applications*, 229:120486. DOI: 10.1016/j.eswa.2023.120486.

Abbadini, M., Beretta, M., Facchinetti, D., Oldani, G., Rossi, M., and Paraboschi, S. (2023). Lightweight cloud application sandboxing. In *2023 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 139–146. IEEE. DOI: 10.1109/cloudcom59040.2023.00033.

Abdulah, S., Ejarque, J., Marzouk, O., Ltaief, H., Sun, Y., Genton, M. G., Badia, R. M., and Keyes, D. E. (2024). Portability and scalability evaluation of large-scale statistical modeling and prediction software through hpc-ready containers. *Future Generation Computer Systems*, 161:248–258. DOI: 10.1016/j.future.2024.06.057.

Agrawal, S. and Singh, D. (2024). Study containerization technologies like docker and kubernetes and their role in modern cloud deployments. In *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, pages 1–5. IEEE. DOI: 10.1109/i2ct61223.2024.10543986.

Ajith, V., Cyriac, T., Chavda, C., Kiyani, A. T., Chennareddy, V., and Ali, K. (2024). Analyzing docker vulnerabilities through static and dynamic methods and enhancing iot security with aws iot core, cloudwatch, and guardduty. *IoT*, 5(3):592–607. DOI: 10.3390/iot5030026.

Aktolga, I. T., Kuru, E. S., Sever, Y., and Angin, P. (2023). Ai-driven container security approaches for 5g and beyond: A survey. *arXiv preprint arXiv:2302.13865*. DOI: 10.52953/zrck3746.

Al-Obaidi, Y., Sorokos, I., and Schmidt, A. (2023). Safetykube: Towards orchestration at the edge for critical production systems. DOI: 10.3850/978-981-18-8071-$1_p217-cd$.

Al Qausar, M. J., Soeparno, H., Gaol, F. L., and Arifin, Y. (2023). Software metrics for container-based applications: Systematic literature review. In *2023 International Conference on Information Management and Technology (ICIMTech)*, pages 125–130. IEEE. DOI: 10.1109/icimtech59029.2023.10277948.

Alamoush, A. and Eichelberger, H. (2024). Open source container orchestration for industry 4.0–requirements and systematic feature analysis. *International Journal on Software Tools for Technology Transfer*, 26(5):527–550. DOI: 10.1007/s10009-024-00767-w.

Aldiabat, M., Yaseen, Q. M., and Ein, Q. A. (2024). An efficient random forest classifier for detecting malicious docker images in docker hub repository. *IEEE Access*. DOI: 10.1109/access.2024.3506663.

Alelyani, A., Datta, A., and Hassan, G. M. (2023). Dascheduler: Dependency-aware scheduling algorithm for containerized dependent jobs. *Journal of Grid Computing*, 21(3):46. DOI: 10.1007/s10723-023-09679-6.

Ali, A. Q., Sultan, A. B. M., Ghani, A. A. A., and Zulzalil, H. (2019). A systematic mapping study on the customization solutions of software as a service applications. *IEEE Access*, 7:88196–88217. DOI: 10.1109/ACCESS.2019.2925499.

Alif, R. G. and Munggaran, L. C. (2024). Implementation of gitops in containerized infrastructure. *Rabit: Jurnal Teknologi dan Sistem Informasi Univrab*, 9(1):154–161. DOI: 10.36341/rabit.v9i1.3787.

Almanasreh, E., Moles, R., and Chen, T. F. (2019). Evaluation of methods used for estimating content validity. *Research in social and administrative pharmacy*, 15(2):214–221. DOI: 10.1016/j.sapharm.2018.03.066.

Almoudane, M. (2025). A comprehensive study on virtualization techniques and their role in cloud computing. DOI: 10.13140/RG.2.2.14067.64800.

Álvarez, G. C., Ramírez, C. A. A., Andión, J., Dueñas, J. C., *et al*. (2023). Toward an integrated and supported machine learning process. In *2023 7th International Young Engineers Forum (YEF-ECE)*, pages 37–42. IEEE. DOI: 10.1109/YEF-ECE58420.2023.10209330.

Alyas, T., Tabassum, N., Iqbal, M. W., Alshahrani, A. S., Alghamdi, A., and Shahzad, S. K. (2023). Resource based automatic calibration system (rbacs) using kubernetes framework. *Intelligent Automation & Soft Computing*, 35(1). DOI: 10.32604/iasc.2023.028815.

Amoiridis, V., Behrens, U., Bocci, A., Branson, J., Brummer, P., Cano, E., Cittolin, S., Da Quintanilha, J. D. S. A., Darlea, G.-L., Deldicque, C., *et al*. (2024). Towards a container-based architecture for cms data acquisition. In *EPJ Web of Conferences*, volume 295, page 02031. EDP Sciences. DOI: 10.1051/epjconf/202429502031.

Antonova, V., Egorov, M., Blinov, V., Malikova, E., and Malikov, A. (2024). Studying the principles of infocommunication network virtualisation using the docker platform. In *2024 Systems of Signals Generating and Processing in the Field of on Board Communications*, pages 1–5. IEEE. DOI: 10.1109/ieeeconf60226.2024.10496739.

Arifiansyah, F. *et al*. (2023). Utilization of virtual reality for docker learning. In *2023 IEEE International Conference on Data and Software Engineering (ICoDSE)*, pages 244–249. IEEE. DOI: 10.1109/icodse59534.2023.10291318.

Ashari, M. F., Bhawiyuga, A., and Basuki, A. (2023). The development of hands-on lab platform using container-based virtualization technology. In *Proceedings of the 8th International Conference on Sustainable Information Engineering and Technology*, pages 304–310. DOI: 10.1145/3626641.3626937.

Augustyn, D. R., Wyciślik, Ł., and Sojka, M. (2024). Tuning a kubernetes horizontal pod autoscaler for meeting performance and load demands in cloud deployments. *Applied Sciences*, 14(2):646. DOI: 10.3390/app14020646.

Aung, L. H., Aung, S. T., Funabiki, N., Kyaw, H. H. S., and Kao, W.-C. (2024). An implementation of web-based answer platform in the flutter programming learning assistant system using docker compose. *Electronics*, 13(24):4878. DOI: 10.3390/electronics13244878.

Azuma, H., Matsumoto, S., Kamei, Y., and Kusumoto, S. (2022). An empirical study on self-admitted technical debt in dockerfiles. *Empirical Software Engineering*, 27(2):49. DOI: 10.1007/s10664-021-10081-7.

Bannon, R. (2022). *Leveraging machine learning to reduce cold start latency of containers in serverless computing*. PhD thesis, Dublin, National College of Ireland. Avaialble at:https://norma.ncirl.ie/5971/.

Barbie, A., Hasselbring, W., and Hansen, M. (2023). Enabling automated integration testing of smart farming applications via digital twin prototypes. In *2023 IEEE Smart World Congress (SWC)*, pages 1–8. IEEE. DOI: 10.1109/swc57546.2023.10449240.

Baresi, L., Quattrocchi, G., and Rasi, N. (2024a). A qualitative and quantitative analysis of container engines. *Journal of Systems and Software*, 210:111965. DOI: 10.1016/j.jss.2024.111965.

Baresi, L., Quattrocchi, G., and Rasi, N. (2024b). A qualitative and quantitative analysis of container engines. *Journal of Systems and Software*, 210:111965. DOI: 10.1016/j.jss.2024.111965.

Barletta, M., Cinque, M., Di Martino, C., Kalbarczyk, Z. T., and Iyer, R. K. (2024a). Mutiny! how does kubernetes fail, and what can we do about it? In *2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 1–14. IEEE. DOI: 10.1109/dsn58291.2024.00016.

Barletta, M., De Simone, L., Della Corte, R., and Di Martino, C. (2024b). Failover timing analysis in orchestrating container-based critical applications. In *2024 19th European Dependable Computing Conference (EDCC)*, pages 81–84. IEEE. DOI: 10.1109/edcc61798.2024.00026.

Barnawi, N., AITooq, R., and Almukaynizi, M. (2024). Mitigating container escape threats through effective countermeasures: A survey. In *Proceedings of the 10th World Congress on Electrical Engineering and Computer Systems and Sciences (EECSS'24)*. DOI: 10.11159/cist24.164.

Barros, P. H., Oliveira, M. Q., Orang, O., da Silva, F. A., Erazo-Costa, F. J., Bastos, A. T., Silva, P. C., dos Santos, G. S., Loureiro, A. A., Ravetti, M. G., *et al.* (2024). Flautim: A federated learning platform using k8s and flower. In *Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*, pages 87–90. SBC. DOI: 10.5753/webmedia$_e stendido$.2024.244009.

Bellavista, P., Dahdal, S., Foschini, L., Tazzioli, D., Tortonesi, M., and Venanzi, R. (2024). Kubernetes enhanced stateful service migration for ml-driven applications in industry 4.0 scenarios. In *2024 IEEE Annual Congress on Artificial Intelligence of Things (AIoT)*, pages 25–31. IEEE. DOI: 10.1109/aiot63253.2024.00015.

Bentaleb, O., Belloum, A. S., Sebaa, A., and El-Maouhab, A. (2022a). Containerization technologies: taxonomies, applications and challenges. *Journal of Supercomputing*, 78(1):1144–1181. DOI: 10.1007/s11227-021-03914-1.

Bentaleb, O., Belloum, A. S., Sebaa, A., and El-Maouhab, A. (2022b). Containerization technologies: Taxonomies, applications and challenges. *The Journal of Supercomputing*, 78(1):1144–1181. DOI: 10.1007/s11227-021-03914-1.

Bentaleb, O., Sebaa, A., Kalli, S., and Belloum, A. S. (2022c). Deployment of a programming framework based on microservices and containers with application to the astrophysical domain. *Astronomy and Computing*, 41:100655. DOI: 10.1016/j.ascom.2022.100655.

Benzi, M., Lagorio, G., and Ribaudo, M. (2022). Automatic challenge generation for hands-on cybersecurity training. In *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 496–503. IEEE. DOI: 10.1109/eurospw55150.2022.00059.

Betz, T., Wen, L., Pan, F., Kaljavesi, G., Zuepke, A., Bastoni, A., Caccamo, M., Knoll, A., and Betz, J. (2024). A containerized microservice architecture for a ros 2 autonomous driving software: An end-to-end latency evaluation. In *2024 IEEE 30th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 57–66. IEEE. DOI: 10.1109/rtcsa62462.2024.00018.

Bhuiyan, M. O. H., Das, S., Majumder, S. H., Majumdar, S., and Hossain, M. S. (2024). On detecting malicious code injection by monitoring multi-level container activities. In *CLOSER*, pages 15–26. DOI: 10.5220/0012509400003711.

Blanco, D. F., Le Mouël, F., Lin, T., and Rekik, A. (2023). Can software containerisation fit the car on-board systems? In *2023 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 123–130. IEEE. DOI: 10.1109/cloudcom59040.2023.00031.

Botez, R., Petruti, C.-M., Ivanciu, I.-A., and Dobrota, V. (2022). Kubernetes-based load balancer as a service for private cloud infrastructures. In *2022 14th International Conference on Communications (COMM)*, pages 1–6. IEEE. DOI: 10.1109/comm54429.2022.9817323.

Bracke, V., Santos, J., Wauters, T., De Turck, F., and Volckaert, B. (2024). A multiobjective metaheuristic-based container consolidation model for cloud application performance improvement. *Journal of Network and Systems Management*, 32(3):61. DOI: 10.1007/s10922-024-09835-7.

Bracke, V., Werrebrouck, G., Santos, J., Wauters, T., De Turck, F., and Volckaert, B. (2023). Online dynamic container rescheduling for improved application service time. *Journal of Network and Systems Management*, 31(4):80. DOI: 10.1007/s10922-023-09766-9.

Burchart, M. and Haake, J. M. (2024). Supporting collaborative writing tasks in large-scale distance education. *IEEE Transactions on Learning Technologies*, 17:1051–1068. DOI: 10.1109/tlt.2024.3355791.

Candela-Uribe, C., Sepúlveda-Rodríguez, L., Chavarro-Porras, J., Sanabria-Ordoñez, J., Garrido, J., Rodríguez-Domínguez, C., and Guerrero-Contreras, G. (2020). SMS-Builder Project. Available at:https://github.com/grid-uq/sms-builder. Accessed: 2025-08-11.

Candela-Uribe, C. A., Sepúlveda-Rodríguez, L. E., Chavarro-Porras, J. C., Sanabria-Ordoñez, J. A., Garrido, J. L., Rodríguez-Domínguez, C., and Guerrero-Contreras, G. (2022). SMS-Builder: An adaptive software tool for building systematic mapping studies. *SoftwareX*, 17:100935. DOI: 10.1016/j.softx.2021.100935.

Candemir, A. İ. and İncereis, N. (2024). Comparative analysis of transaction performance in different virtualization environments. *Acta Infologica*, 8(2):176–187. DOI: 10.26650/acin.1462076.

Carrión, C. (2022a). Kubernetes as a standard container orchestrator-a bibliometric analysis. *Journal of Grid Computing*, 20(4):42. DOI: 10.1007/s10723-022-09629-8.

Carrión, C. (2022b). Kubernetes scheduling: Taxonomy, ongoing issues and challenges. *ACM Computing Surveys*, 55(7):1–37. DOI: 10.1145/3539606.

Chamoli, S. and Mittal, V. (2023). Docker networking: A security review. In *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 624–629. IEEE. DOI: 10.1109/icoei56765.2023.10126070.

Chaurasia, P., Nath, S. B., Addya, S. K., and Ghosh, S. K. (2022). Automating the selection of container orchestrators for service deployment. In *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pages 739–743. IEEE. DOI: 10.1109/comsnets53615.2022.9668432.

Chen, C., Shen, W., and Liu, H. (2022). Construction of cloud computing password application system based on docker technology. In *2022 International Conference on Artificial Intelligence of Things and Crowdsensing (AIoTCs)*, pages 175–179. IEEE. DOI: 10.1109/aiotcs58181.2022.00033.

Chen, K., Zhao, Y., Guo, J., Gu, Z., Han, L., and Tang, K. (2024). A container escape detection method based on a dependency graph. *Electronics*, 13(23):4773. DOI: 10.3390/electronics13234773.

Choi, D., Seo, H., Kim, K., You, M., Shin, S., and Kim, J. (2024). Uncovering threats in container systems: A study on misconfigured container components in the wild. *IEEE Access*. DOI: 10.1109/access.2024.3514751.

Choi, K., Seo, H., Han, H., Ryu, M., and Kang, S. (2023a). Credscache: Making overlayfs scalable for containerized services. *Future Generation Computer Systems*, 147:44–58. DOI: 10.1016/j.future.2023.04.027.

Choi, Y.-D., Roy, B., Nguyen, J., Ahmad, R., Maghami, I., Nassar, A., Li, Z., Castronova, A. M., Malik, T., Wang, S., *et al.* (2023b). Comparing containerization-based approaches for reproducible computational modeling of environmental systems. *Environmental Modelling & Software*, 167:105760. DOI: 10.1016/j.envsoft.2023.105760.

Clement, M. (2025). Role of cloud-native technologies in agile software developmen. Avaialble at:`https://dspace.mit.edu/entities/publication/f5efb850-6c68-4543-a910-c3c9690b332a`.

Dakić, V., Kovač, M., and Slovinac, J. (2024). Evolving high-performance computing data centers with kubernetes, performance analysis, and dynamic workload placement based on machine learning scheduling. *Electronics*, 13(13):2651. DOI: 10.3390/electronics13132651.

de Oliveira Filho, A. T., Freitas, E., do Carmo, P. R., Souto, E., Kelner, J., and Sadok, D. F. (2024). Analysis of sr-iov in docker containers using rtt measurements. *Computer Communications*, 228:107961. DOI: 10.1016/j.comcom.2024.107961.

Deng, Q., Tan, X., Yang, J., Zheng, C., Wang, L., and Xu, Z. (2022). A secure container placement strategy using deep reinforcement learning in cloud. In *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 1299–1304. IEEE. DOI: 10.1109/cscwd54268.2022.9776226.

Dimova, T., Kalendar, I., Denkovski, D., Efnusheva, D., and Kalendar, M. (2024). An innovative approach of api automation testing implemented on cloud environments using container management services. Available at:`https://icaiit.org/paper.php?paper=12th_ICAIIT_1/4_1`.

Dipta, D. R., Tiemann, T., Gulmezoglu, B., Marin, E., and Eisenbarth, T. (2024). Dynamic frequency-based fingerprinting attacks against modern sandbox environments. In *2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P)*, pages 327–344. IEEE. DOI: 10.1109/eurosp60621.2024.00025.

Dobslaw, F., Angelin, K., Öberg, L.-M., and Ahmad, A. (2023). The gap between higher education and the software industry—a case study on technology differences. In *Proceedings of the 5th European Conference on Software Engineering Education*, pages 11–21. DOI: 10.1145/3593663.3593690.

Dogani, J., Khunjush, F., and Seydali, M. (2022). K-agrued: A container autoscaling technique for cloud-based web applications in kubernetes using attention-based gru encoder-decoder. *Journal of Grid Computing*, 20(4):40. DOI: 10.1007/s10723-022-09634-x.

Dogani, J., Namvar, R., and Khunjush, F. (2023). Auto-scaling techniques in container-based cloud and edge/fog computing: Taxonomy and survey. *Computer Communications*, 209:120–150. DOI: 10.1016/j.comcom.2023.06.010.

Du, W., Zeng, H., and Won, K. (2022). Seed emulator: An internet emulator for research and education. In *Proceedings of the 21st ACM workshop on hot topics in networks*, pages 101–107. DOI: 10.1145/3563766.3564097.

Dubey, A., Singh, C. P., and Nadig, D. (2024). Leveraging large language models for intent-based generation of cloud-native configurations. In *2024 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6. IEEE. DOI: 10.1109/ants63515.2024.10898244.

Ebrahimpour, H., Ashtiani, M., Bakhshi, F., and Bakhtiari-azad, G. (2023). A heuristic-based package-aware function scheduling approach for creating a trade-off between cold start time and cost in faas computing environments. *The Journal of Supercomputing*, 79(11):12142–12190. DOI: 10.1007/s11227-023-05128-z.

El Khairi, A., Caselli, M., Knierim, C., Peter, A., and Continella, A. (2022). Contextualizing system calls in containers for anomaly-based intrusion detection. In *Proceedings of the 2022 on Cloud Computing Security Workshop*, pages 9–21. DOI: 10.1145/3560810.3564266.

Eng, K., Hindle, A., and Stroulia, E. (2024). Patterns of multi-container composition for service orchestration with docker compose. *Empirical Software Engineering*, 29(3):65. DOI: 10.1007/s10664-024-10462-8.

Entrialgo, J., García, M., García, J., López, J. M., and Díaz, J. L. (2024). Joint autoscaling of containers and virtual machines for cost optimization in container clusters. *Journal of Grid Computing*, 22(1):17. DOI: 10.1007/s10723-023-09732-4.

Eroshkin, I., Vojtech, L., and Neruda, M. (2024). Improving quality of 5g/6g/xg network through reliability estimation in kubernetes. In *European Wireless 2024; 29th European Wireless Conference*, pages 77–84. VDE. Available at:`https://ieeexplore.ieee.org/document/10925937`.

Ersted Rasmussen, M., Dueholm Vestergaard, C., Folsted Kallehauge, J., Ren, J., Haislund Guldberg, M., Nørrevang, O., *et al.* (2024). Raddeploy: A framework for integrating in-house developed software and artificial intelligence models seamlessly into radiotherapy workflows. phys imaging radiat oncol 2024; 31.

Fava, F. B., Leite, L. F. L., Da Silva, L. F. A., Costa, P. R. D. S. A., Nogueira, A. G. D., Lopes, A. F. G., Schepke, C., Kreutz, D. L., and Mansilha, R. B. (2024). Assessing the performance of docker in docker containers for microservice-based architectures. In *2024 32nd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 137–142. IEEE. DOI: 10.1109/pdp62718.2024.00026.

Fernalld, K., OConnor, T., Sudhakaran, S., and Nur, N. (2023). Lightweight symphony: Towards reducing computer science student anxiety with standardized docker environments. In *Proceedings of the 24th Annual Conference on Information Technology Education*, pages 15–21. DOI: 10.1145/3585059.3611432.

Fischer, T., Hirn, D., and Kul, G. (2024). A reproducible tutorial on reproducibility in database systems research. *Proceedings of the VLDB Endowment*, 17(12):4221–4224. DOI: 10.14778/3685800.3685840.

Flora, J. and Antunes, N. (2024). Evaluating intrusion detection for microservice applications: Benchmark, dataset, and case studies. *Journal of Systems and Software*, 216:112142. DOI: 10.1016/j.jss.2024.112142.

Frasão, A., Heinrich, T., Fulber-Garcia, V., Will, N. C., Obelheiro, R. R., and Maziero, C. A. (2024). I see syscalls by the seashore: An anomaly-based ids for containers leveraging sysdig data. In *2024 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6. IEEE. DOI: 10.1109/ISCC61673.2024.10733595.

Fu, Y., Machlovi, N., Mao, Y., Wang, J., Cheng, L., and Liu, Q. (2022). Performance evaluation of resource management schemes for cloud native platforms with computing containers. In *2022 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pages 414–415. IEEE. DOI: 10.1109/ipccc55026.2022.9894300.

Gackstatter, P., Frangoudis, P. A., and Dustdar, S. (2022). Pushing serverless to the edge with webassembly runtimes. In *2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 140–149. IEEE. DOI: 10.1109/ccgrid54584.2022.00023.

Galantino, S., Risso, F., Cazzaniga, A., Garrone, F., Terruggia, R., and Lazzari, R. (2022). An edge-based architecture for phasor measurements in smart grids. In *2022 AEIT International Annual Conference (AEIT)*, pages 1–6. IEEE. DOI: 10.23919/aeit56783.2022.9951842.

Gamess, E. and Parajuli, M. (2024). Image-processing workloads and ddos attack resilience: Evaluating docker and podman containers on raspberry pi and odroid. In *Proceedings of the 2024 ACM Southeast Conference*, pages 138–147. DOI: 10.1145/3603287.3651219.

Ganne, A. (2022). Cloud data security methods: Kubernetes vs docker swarm. *International Research Journal of Modernization in Engineering Technology*, 4(11):1–6. DOI: 10.56726/irjmets32176.

Gao, P., Wan, R., Wang, C., and Cheng, Y. (2022). Container load prediction algorithm based on arima model and bp neural network. In *2022 IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 334–337. IEEE. DOI: 10.1109/iaeac54830.2022.9929476.

Geetha, J., Jayalakshmi, D., Naresh, E., and Sreenivasa, N. (2024). Lightweight cloud-based solution for digital education and assessment. *Science & Technology Libraries*, 43(3):274–286. DOI: 10.1080/0194262x.2023.2247460.

Geng, J., Zheng, X., Lai, X., Zhao, L., and Xu, Y. (2023). Construction and implementation of electric power deep learning computing service platform. In *2023 4th International Conference on Computer Engineering and Intelligent Control (ICCEIC)*, pages 492–495. IEEE. DOI: 10.1109/icceic60201.2023.10426645.

Gharaibeh, T., Seiden, S., Abouelsaoud, M., Bou-Harb, E., and Baggili, I. (2024). Don't, stop, drop, pause: Forensics of container checkpoints (conpoint). In *Proceedings of the 19th International Conference on Availability, Reliability and Security*, pages 1–11. DOI: 10.1145/3664476.3670895.

Ghorbian, M. and Ghobaei-Arani, M. (2024). A survey on the cold start latency approaches in serverless computing: an optimization-based perspective. *Computing*, 106(11):3755–3809. DOI: 10.1007/s00607-024-01335-5.

González-Abad, J., Lopez Garcia, A., and Kozlov, V. Y. (2023). A container-based workflow for distributed training of deep learning algorithms in hpc clusters. *Cluster Computing*, 26(5):2815–2834. DOI: 10.1007/s10586-022-03798-7.

Goodman, L. A. (1961). Snowball sampling. *The annals of mathematical statistics*, pages 148–170. DOI: 10.1214/aoms/1177705148.

Gu, Y., Tan, X., Zhang, Y., Gao, S., and Yang, M. (2025). Epscan: Automated detection of excessive rbac permissions in kubernetes applications. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 3199–3217. IEEE. DOI: 10.1109/sp61157.2025.00011.

Gupta, C., Van Ede, T., and Continella, A. (2023). Honeykube: Designing and deploying a microservices-based web honeypot. In *2023 IEEE Security and Privacy Workshops (SPW)*, pages 1–11. IEEE. DOI: 10.1109/spw59333.2023.00005.

Han, S., Li, W., Zhang, E., Shi, J., Wang, W., and Lu, X. (2022). Mladder: An online training system for machine learning and data science education. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4862–4866. DOI: 10.1145/3511808.3557201.

Hao, Y., Zhang, X., and Wang, D. (2024). Cpced: a container escape detection system based on cni plugin. In *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 242–253. IEEE. DOI: 10.1109/trustcom63139.2024.00057.

Haq, M. S., Tosun, A. Ş., and Korkmaz, T. (2022). Security analysis of docker containers for arm architecture. In *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*, pages 224–236. IEEE. DOI: 10.1109/sec54971.2022.00025.

Haq, M. S., Tosun, A. Ş., and Korkmaz, T. (2025). Lucid:

A framework for reducing false positives and inconsistencies among container scanning tools. *IEEE Access*. DOI: 10.1109/access.2025.3587626.

Haresh, S. *et al.* (2023). Virtualization using docker container for reproducible desktop environment. In *2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS)*, pages 1–4. IEEE. DOI: 10.1109/iccebs58601.2023.10448590.

Hassan, A., Bhatti, S. H., Shujaat, S., and Hwang, Y. (2022). To adopt or not to adopt? the determinants of cloud computing adoption in information technology sector. *Decision Analytics Journal*, 5:100138. DOI: 10.1016/j.dajour.2022.100138.

Hettiarachchi, L. S., Jayadeva, S. V., Bandara, R. A. V., Palliyaguruge, D., Arachchillage, U. S. S. S., and Kasthurirathna, D. (2022). Artificial intelligence-based centralized resource management application for distributed systems. In *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–6. IEEE. DOI: 10.1109/icccnt54827.2022.9984530.

Horchulhack, P., Viegas, E. K., Santin, A. O., Ramos, F. V., and Tedeschi, P. (2024). Detection of quality of service degradation on multi-tenant containerized services. *Journal of Network and Computer Applications*, 224:103839. DOI: 10.1016/j.jnca.2024.103839.

Hristev, R., Veselinova, M., and Kolev, K. (2023). System architecture for automated backup and recovery of disk volumes in containers. Avaialbel at:`https://www.researchgate.net/publication/376750215_SYSTEM_ARCHITECTURE_FOR_AUTOMATED_BACKUP_AND_RECOVERY_OF_DISK_VOLUMES_IN_CONTAINERS`.

Husain, H., Marzuki, K., Mardedi, L. Z. A., *et al.* (2023). Analysis and implementation of comparison between podman and docker in container management. *International Journal of Electronics and Communications Systems*, 3(2):57–67. DOI: 10.24042/ijecs.v3i2.19860.

Jackson, S. and Wurst, K. R. (2022). Teaching with vs code devcontainers: Conference workshop. *Journal of Computing Sciences in Colleges*, 37(8):81–82. Available at:`https://dl.acm.org/doi/abs/10.5555/3533747.3533758`.

Jalali, S. and Wohlin, C. (2012). Systematic literature studies: database searches vs. backward snowballing. In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*, pages 29–38. DOI: 10.1145/2372251.2372257.

Jang, H.-C. and Luo, S.-Y. (2023). Enhancing node fault tolerance through high-availability clusters in kubernetes. In *2023 IEEE 3rd International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB)*, pages 30–35. IEEE. DOI: 10.1109/iceib57887.2023.10170110.

Jeon, J., Park, S., Jeong, B., and Jeong, Y.-S. (2024). Efficient container scheduling with hybrid deep learning model for improved service reliability in cloud computing. *IEEE Access*, 12:65166–65177. DOI: 10.1109/access.2024.3396652.

Jeong, B., Jeon, J., and Jeong, Y.-S. (2023). Proactive resource autoscaling scheme based on scinet for high-performance cloud computing. *IEEE Transactions on Cloud Computing*, 11(4):3497–3509. DOI: 10.1109/tcc.2023.3292378.

Johansson, B., Rågberger, M., Nolte, T., and Papadopoulos, A. V. (2022). Kubernetes orchestration of high availability distributed control systems. In *2022 IEEE International Conference on Industrial Technology (ICIT)*, pages 1–8. IEEE. DOI: 10.1109/icit48603.2022.10002757.

Jolak, R., Rosenstatter, T., Mohamad, M., Strandberg, K., Sangchoolie, B., Nowdehi, N., and Scandariato, R. (2022). Conserve: A framework for the selection of techniques for monitoring containers security. *Journal of Systems and Software*, 186:111158. DOI: 10.1016/j.jss.2021.111158.

Joraviya, N., Gohil, B. N., and Rao, U. P. (2024a). Abhids: An anomaly-based host intrusion detection system using frequency of n-gram system call features and ensemble learning for containerized environment. *Concurrency and Computation: Practice and Experience*, 36(23):e8249. DOI: 10.1002/cpe.8249.

Joraviya, N., Gohil, B. N., and Rao, U. P. (2024b). Dlhids: deep learning-based host intrusion detection system using system calls-to-image for containerized cloud environment: N. joraviya et al. *The Journal of Supercomputing*, 80(9):12218–12246. Available at:`https://www.springerprofessional.de/en/dl-hids-deep-learning-based-host-intrusion-detection-26720536`.

Joshi, S., Hasan, B., and Brindha, R. (2024). Optimal declarative orchestration of full lifecycle of machine learning models for cloud native. In *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pages 578–582. IEEE. DOI: 10.1109/icaaic60222.2024.10575306.

Kaiser, S., Haq, M. S., Tosun, A. S., and Korkmaz, T. (2022a). Container Technologies for ARM Architecture: A Comprehensive Survey of the State-of-the-Art. *IEEE Access*, 10:84853–84881. DOI: 10.1109/ACCESS.2022.3197151.

Kaiser, S., Haq, M. S., Tosun, A. Ş., and Korkmaz, T. (2022b). Container technologies for arm architecture: A comprehensive survey of the state-of-the-art. *IEEE Access*, 10:84853–84881. DOI: 10.1109/access.2022.3197151.

Kaiser, S., şaman Tosun, A., and Korkmaz, T. (2023a). Benchmarking container technologies on arm-based edge devices. *IEEE access*, 11:107331–107347. DOI: 10.1109/access.2023.3321274.

Kaiser, S., Tosun, A. S., and Korkmaz, T. (2023b). Benchmarking Container Technologies on ARM-Based Edge Devices. *IEEE Access*, 11:107331–107347. DOI: 10.1109/ACCESS.2023.3321274.

Kaiser, S., Tosun, A. S., and Korkmaz, T. (2024). Exploring the viability of unikernels for arm-powered edge computing. *arXiv preprint arXiv:2412.03030*. DOI: 10.48550/arxiv.2412.03030.

Kamieniarz, K. and Mazurczyk, W. (2024). A comparative study on the security of kubernetes deployments. In *2024 International Wireless Communications and Mobile Computing (IWCMC)*, pages 0718–0723. IEEE. DOI: 10.1109/iwcmc61514.2024.10592468.

Kanagachalam, S., Tulkinbekov, K., and Kim, D.-H. (2022). Blosm: Blockchain-based service migration for connected cars in embedded edge environment. *Electronics*, 11(3):341. DOI: 10.3390/electronics11030341.

Karamzadeh, A. and Shameli-Sendi, A. (2024). Reducing cold start delay in serverless computing using lightweight virtual machines. *Journal of Network and Computer Applications*, 232:104030. DOI: 10.1016/j.jnca.2024.104030.

Karmakar, G. and Arri, H. S. (2024). Mathematical approaches to securing kubernetes: Analyzing log data volume in a complex landscape. DOI: 10.56155/978-81-955020-9-7-19.

Karumudi, M. *et al.* (2024). Efficient workload portability and optimized resource utilization using containerization in a multi-cloud environment. In *2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)*, pages 823–828. IEEE. DOI: 10.1109/icdici62993.2024.10810796.

Kaur, T. (2024). Containers in multi-cloud environments: Benefits, challenges, and best practices. Available at:`https://www.researchgate.net/publication/382458769_Containers_in_Multi-Cloud_Environments_Benefits_Challenges_and_Best_Practices`.

Keller Tesser, R. and Borin, E. (2023). Containers in hpc: a survey. *The Journal of Supercomputing*, 79(5):5759–5827. DOI: 10.1007/s11227-022-04848-y.

Kim, B. S., Lee, S. H., Lee, Y. R., Park, Y. H., and Jeong, J. (2022). Design and implementation of cloud docker application architecture based on machine learning in container management for smart manufacturing. *Applied Sciences*, 12(13):6737. DOI: 10.3390/app12136737.

Kim, T., Park, S., and Kim, H. (2023). Why johnny can't use secure docker images: Investigating the usability challenges in using docker image vulnerability scanners through heuristic evaluation. In *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*, pages 669–685. DOI: 10.1145/3607199.3607244.

Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., and Linkman, S. (2010). Systematic literature reviews in software engineering-A tertiary study. *Information and Software Technology*, 52(8):792–805. DOI: 10.1016/j.infsof.2010.03.006.

Kithulwatta, W., Jayasena, K., Kumara, B., and Rathnayaka, R. (2022a). Performance evaluation of docker-based apache and nginx web server. In *2022 3rd International Conference for Emerging Technology (INCET)*, pages 1–6. IEEE. DOI: 10.1109/incet54531.2022.9824303.

Kithulwatta, W. M. C. J. T., Jayasena, K. P. N., Kumara, B. T. G. S., and Rathnayaka, R. M. K. T. (2022b). Integration with docker container technologies for distributed and microservices applications: A state-of-the-art review. *International Journal of Systems and Service-Oriented Engineering*, 12(1). DOI: 10.4018/IJSSOE.297136.

Kjorveziroski, V. and Filiposka, S. (2023a). Webassembly as an enabler for next generation serverless computing. *Journal of Grid Computing*, 21(3):34. DOI: 10.1007/s10723-023-09669-8.

Kjorveziroski, V. and Filiposka, S. (2023b). Webassembly orchestration in the context of serverless computing. *Journal of Network and Systems Management*, 31(3):62. DOI: 10.1007/s10922-023-09753-0.

Kosińska, J., Brotoń, G., and Tobiasz, M. (2024). Knowledge representation of the state of a cloud-native application. *International Journal on Software Tools for Technology Transfer*, 26(1):21–32. DOI: 10.1007/s10009-023-00705-2.

Kostolny, J., Karcolova, V., Vaclavkova, M., and Blahova, L. (2023). Enhancing learning outcomes with interactive courses. In *2023 21st International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pages 324–329. IEEE. DOI: 10.1109/iceta61311.2023.10344313.

Kotenko, I. V., Melnik, M. V., and Abramenko, G. T. (2024). Anomaly detection in container systems: Using histograms of normal processes and an autoencoder. In *2024 IEEE 25th International Conference of Young Professionals in Electron Devices and Materials (EDM)*, pages 1930–1934. IEEE. DOI: 10.1109/edm61683.2024.10615118.

Kozhirbayev, Z. and Sinnott, R. O. (2017). A performance comparison of container-based technologies for the cloud. *Future Generation Computer Systems*, 68:175–182. DOI: 10.1016/j.future.2016.08.025.

Kuity, A. and Peddoju, S. K. (2023). Investigating performance metrics for container-based hpc environments using x86 and openpower systems. *Journal of Cloud Computing*, 12(1):178. DOI: 10.1186/s13677-023-00546-z.

Kumar, B., Verma, A., and Verma, P. (2024). Optimizing resource allocation using proactive scaling with predictive models and custom resources. *Computers and Electrical Engineering*, 118:109419. DOI: 10.1016/j.compeleceng.2024.109419.

Kumar, M. and Kaur, G. (2022). An empirical study of containerized mpi and gui application on hpc in the cloud. In *2022 2nd International Conference on Innovative Sustainable Computational Technologies (CISCT)*, pages 1–6. IEEE. DOI: 10.1109/cisct55310.2022.10046607.

Kunekar, P., Bhavsar, N., Das, K., and Bhavsar, S. (2024). Use of docker containerization and load balancer to scale a flask application. In *2024 4th International Conference on Soft Computing for Security Applications (ICSCSA)*, pages 534–541. DOI: 10.1109/ICSCSA64454.2024.00092.

Kurniawan, A. P., Ashar, M. N., Hidayat, F., Salsabila, S., Gautama, P. T. W., Shiddiqi, A. M., and Studiawan, H. (2023). Performance evaluation for deploying dockerized web application on aws, gcp, and azure. In *2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT)*, pages 346–350. IEEE. DOI: 10.1109/iccect57938.2023.10140775.

Kwon, S., Son, W., and Lee, J.-H. (2023). Vdirs: Vulnerable docker image response system. In *Proceedings of the 2023 International Conference on Intelligent Computing and Its Emerging Applications*, pages 78–83. DOI: 10.1145/3659154.3659174.

Lee, H.-K., Han, S.-H., and Lee, D. (2022). Kernel-based container file access control architecture to protect important application information. *Electronics*, 12(1):52. DOI: 10.3390/electronics12010052.

Lee, S. W., Choi, Y. G., and Jeon, J. W. (2024). Enhancing autonomous driving systems through ros2 and aws cloud:

V2i interaction and hpc data processing. In *2024 IEEE 33rd International Symposium on Industrial Electronics (ISIE)*, pages 1–6. IEEE. DOI: 10.1109/isie54533.2024.10595834.

Lee, W., Choe, Y. R., and Ghosh, R. S. (2023). Recurrent neural network and convolutional neural network for detection of denial of service attack in microservices. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, pages 1451–1456. IEEE. DOI: 10.1109/icmla58977.2023.00219.

Li, B., Zhan, Y., and Ren, S. (2023a). A fast cold-start solution: container space reuse based on resource isolation. *Electronics*, 12(11):2515. DOI: 10.3390/electronics12112515.

Li, G., Takahashi, K., Ichikawa, K., Iida, H., Nakasan, C., Leelaprute, P., Thiengburanathum, P., and Phannachitta, P. (2023b). The convergence of container and traditional virtualization: Strengths and limitations. *SN Computer Science*, 4(4):387. DOI: 10.1007/s42979-023-01827-9.

Li, M., Hu, H., and Liu, W. (2024a). A secure container placement algorithm based on microservice invocation criticality. In *2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT)*, pages 234–240. IEEE. DOI: 10.1109/iccect60629.2024.10545879.

Li, Z., Zeng, D., and Chen, R. (2024b). Webassembly or container? joint optimization of microservice consolidation and deployment towards cost efficient edge-end consortium. In *2024 IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS)*, pages 1–10. IEEE. DOI: 10.1109/iwqos61813.2024.10682874.

Liagkou, V., Fragiadakis, G., Filiopoulou, E., Michalakelis, C., Kamalakis, T., and Nikolaidou, M. (2022). A pricing model for container-as-a-service, based on hedonic indices. *Simulation Modelling Practice and Theory*, 115:102441. DOI: 10.1016/j.simpat.2021.102441.

Ma, T., Luo, L., Yu, H., Chen, X., Xie, J., Ma, C., Xie, Y., Sun, G., Wei, T., Chen, L., *et al.* (2024). Klonet: an $easy-to-use$ and scalable platform for computer networks education. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pages 2025–2046. Available at:https://www.usenix.org/conference/nsdi24/presentation/ma.

Madi, T. and Esteves-Verissimo, P. (2022). A fault and intrusion tolerance framework for containerized environments: A specification-based error detection approach. In *2022 International Workshop on Secure and Reliable Microservices and Containers (SRMC)*, pages 1–8. IEEE. DOI: 10.1109/srmc57347.2022.00005.

Mailewa, A., Mengel, S., Gittner, L., and Khan, H. (2022). Mechanisms and techniques to enhance the security of big data analytic framework with mongodb and linux containers. *Array*, 15:100236. DOI: 10.2139/ssrn.3990567.

Malan, D. J. (2024). Containerizing cs50: Standardizing students' programming environments. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, pages 534–540. DOI: 10.1145/3649217.3653567.

Malhotra, R., Bansal, A., and Kessentini, M. (2023). Vulnerability analysis of docker hub official images and verified images. In *2023 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pages 150–155. IEEE. DOI: 10.1109/sose58276.2023.00025.

Malhotra, R., Bansal, A., and Kessentini, M. (2024a). A Systematic Literature Review on Maintenance of Software Containers. *ACM Computing Surveys*, 56(8). DOI: 10.1145/3645092.

Malhotra, R., Bansal, A., and Kessentini, M. (2024b). A systematic literature review on maintenance of software containers. *ACM Computing Surveys*, 56(8):1–38. DOI: 10.1145/3645092.

Malviya, A. and Dwivedi, R. K. (2022). A comparative analysis of container orchestration tools in cloud computing. In *2022 9th International conference on computing for sustainable global development (INDIACom)*, pages 698–703. IEEE. DOI: 10.23919/indiacom54597.2022.9763171.

Mehran, A. and Ulus, D. (2024). Runtime verification containers for publish/subscribe networks. *arXiv preprint arXiv:2408.06380*. DOI: 10.48550/arxiv.2408.06380.

Melo, P., Gama, L., Dantas, J., Beserra, D., and Araujo, J. (2023). Performance evaluation of container management tasks in os-level virtualization platforms. In *2023 IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 1–6. IEEE. DOI: 10.1109/wetice57085.2023.10477783.

Mills, A., White, J., and Legg, P. (2023). Longitudinal risk-based security assessment of docker software container images. *Computers & Security*, 135:103478. DOI: 10.1016/j.cose.2023.103478.

Modey, P., Freeman, E., Opoku-Yeboah, S., and Avulekpor, E. (2024). A hybrid security algorithm in container-based virtualization environments using residue number systems. In *2024 Conference on Information Communications Technology and Society (ICTAS)*, pages 107–112. IEEE. DOI: 10.1109/ictas59620.2024.10507123.

Molnár, G., József, C., and Éva, K. (2023). Evaluation and technological solutions for a dynamic, unified cloud programming development environment: Ease of use and applicable system for uniformized practices and assessments. In *2023 IEEE 21st World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 000237–000240. IEEE. DOI: 10.1109/SAMI58000.2023.10044519.

Mondal, S. K., Wu, X., and Pan, R. (2024a). On rapid application deployment with kubernetes. In *Proceedings of the 2024 4th International Conference on Artificial Intelligence, Automation and Algorithms*, pages 76–80. DOI: 10.1145/3700523.3700538.

Mondal, S. K., Zheng, Z., and Cheng, Y. (2024b). On the optimization of kubernetes toward the enhancement of cloud computing. *Mathematics*, 12(16):2476. DOI: 10.3390/math12162476.

Moric, Z., Dakic, V., and Kulic, M. (2024). Implementing a security framework for container orchestration. In *2024 IEEE 11th International Conference on Cyber Security and Cloud Computing (CSCloud)*, pages 200–206. IEEE. DOI: 10.1109/cscloud62866.2024.00042.

Mourao, E., Kalinowski, M., Murta, L., Mendes, E., and Wohlin, C. (2017). Investigating the Use of a Hybrid Search Strategy for Systematic Reviews. *International Sympo-*

sium on Empirical Software Engineering and Measurement, 2017-November:193–198. DOI: 10.1109/ESEM.2017.30.

Moussa, W., Nashaat, M., Saber, W., and Rizk, R. (2022). Comprehensive study on machine learning-based container scheduling in cloud. In *International Conference on Advanced Machine Learning Technologies and Applications*, pages 581–592. Springer. DOI: 10.1007/978-3-031-03918-$8_4$8.

Moysiadis, V., Tsakos, K., Sarigiannidis, P., Petrakis, E. G., Boursianis, A. D., and Goudos, S. K. (2022). A cloud computing web-based application for smart farming based on microservices architecture. In *2022 11th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, pages 1–5. IEEE. DOI: 10.1109/mocast54814.2022.9837727.

Mthembu, S. J., Ezeji, I. N., and Adigun, M. (2023). Orchestration tools for efficient deployment of iot applications in fog computing: A systematic review. In *International Conference on Artificial Intelligence and its Applications*, pages 146–151. DOI: 10.59200/icarti.2023.021.

Mujkanovic, N., Durillo, J. J., Hammer, N., and Müller, T. (2023). Survey of adaptive containerization architectures for hpc. In *Proceedings of the SC'23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis*, pages 165–176. DOI: 10.1145/3624062.3624588.

Nakakaze, O., Koren, I., Brillowski, F., and Klamma, R. (2024). Adaptive retrofitting for industrial machines: utilizing webassembly and peer-to-peer connectivity on the edge. *World Wide Web*, 27(1):7. DOI: 10.1007/s11280-024-01237-8.

Nakarmi, A., Kesharwani, H., Mallick, T., Jhingran, S., and Raj, G. (2024). A comprehensive study on optimization techniques for microservices deployment. In *2024 Sixth International Conference on Computational Intelligence and Communication Technologies (CCICT)*, pages 133–140. IEEE. DOI: 10.1109/ccict62777.2024.00033.

Narasimhulu, M., Mounika, D. V., Varshini, P., Rao, T. R. K., *et al.* (2023). Investigating the impact of containerization on the deployment process in devops. In *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*, pages 679–685. IEEE. DOI: 10.1109/icecaa58104.2023.10212240.

Naydenov, N. and Ruseva, S. (2023). Cloud container orchestration architectures, models and methods: a systematic mapping study. In *2023 22nd International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–8. IEEE. DOI: 10.1109/infoteh57020.2023.10094059.

Ndigande, A. P., Ari, I., and Ozer, S. (2024). Analysis and comparison of dockerized and standalone apache spark configurations for efficient distributed data processing. In *2024 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 1–5. IEEE. DOI: 10.1109/asyu62119.2024.10757047.

Needleman, I. G. (2002). A guide to systematic reviews. *Journal of Clinical Periodontology*, 29(SUPPL. 3):6–9. DOI: 10.1034/j.1600-051x.29.s3.15.x.

Nelson, C. and Shoshitaishvili, Y. (2024). Dojo: applied cybersecurity education in the browser. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, pages 930–936. DOI: 10.1145/3626252.3630836.

Nguyen, P. H., Kramer, M., Klein, J., and Traon, Y. L. (2015). An extensive systematic review on the Model-Driven Development of secure systems. *Information and Software Technology*, 68:62–81. DOI: 10.1016/j.infsof.2015.08.006.

Pankowski, A. and Powroźnik, P. (2023). Comparison of application container orchestration platforms. *Journal of Computer Sciences Institute*, 29:383–390. DOI: 10.35784/jcsi.3823.

Pastor-Galindo, J., Sandlin, H.-Â., Mármol, F. G., Bovet, G., and Pérez, G. M. (2024). A big data architecture for early identification and categorization of dark web sites. *Future Generation Computer Systems*, 157:67–81. DOI: 10.1016/j.future.2024.03.025.

Patra, M. K., Kumari, A., Sahoo, B., and Turuk, A. K. (2022). Docker security: Threat model and best practices to secure a docker container. In *2022 IEEE 2nd International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*, pages 1–6. IEEE. DOI: 10.1109/isssc56467.2022.10051481.

Patra, M. K., Sahoo, B., and Turuk, A. K. (2023). Container as a service in the cloud: An approach to secure hybrid virtualization. In *Recent Trends and Best Practices in Industry 4.0*, pages 59–76. River Publishers. DOI: 10.1201/9781003441717-3.

Pavao, A., Guyon, I., Letournel, A.-C., Tran, D.-T., Baro, X., Escalante, H. J., Escalera, S., Thomas, T., and Xu, Z. (2023). Codalab competitions: An open source platform to organize scientific challenges. *Journal of Machine Learning Research*, 24(198):1–6. Available at:`https://www.jmlr.org/papers/v24/21-1436.html`.

Pérez, R., Rivera, M., Salgueiro, Y., Baier, C. R., and Wheeler, P. (2023). Moving microgrid hierarchical control to an sdn-based kubernetes cluster: A framework for reliable and flexible energy distribution. *Sensors*, 23(7):3395. DOI: 10.3390/s23073395.

Petticrew, M. and Roberts, H. (2008). *Systematic reviews in the social sciences: A practical guide*. John Wiley & Sons. DOI: 10.5860/choice.43-5664.

Purahong, B., Sithiyopasakul, J., Sithiyopasakul, P., Lasakul, A., and Benjangkaprasert, C. (2023). Automated resource management system based upon container orchestration tools comparison. *Journal of Advances in Information Technology*, 14(3). DOI: 10.12720/jait.14.3.501-509.

Qian, C. (2022). An accurate financial aid system for university students built on a cloud platform. In *2022 International Conference on Knowledge Engineering and Communication Systems (ICKES)*, pages 1–6. IEEE. DOI: 10.1109/ickecs56523.2022.10059880.

Raj, P., Kumar, R. S., and Kumar, A. (2022). Cloud security. In *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, pages 1890–1894. IEEE. DOI: 10.1109/icac3n56670.2022.10074139.

Raj, V. (2023). A framework for migration of microservices based applications to serverless platform with efficient cold start latency. *Authorea Preprints*. DOI:

10.36227/techrxiv.24237196.

Rajasekar, V., Saračević, M., Karabačević, D., Stanujkić, D., Hasić, A., Azizović, M., and Thirumalai, S. (2024). Security-enhanced qos-aware autoscaling of kubernetes pods using horizontal pod autoscaler (hpa). *Journal of intelligent management decision*, 3(3):175–186. DOI: 10.56578/jimd030304.

Ramanathan, S., Bhattacharyya, A., Kondepu, K., and Fumagalli, A. (2024). Enabling containerized central unit live migration in 5g radio access network: An experimental study. *Journal of Network and Computer Applications*, 221:103767. DOI: 10.1016/j.jnca.2023.103767.

Rashid, S. F. and Qasha, R. P. (2022). Cloud-based big data approach for managing multi-types data of cultural heritage. In *2022 International Conference on Communications, Information, Electronic and Energy Systems (CIEES)*, pages 1–6. IEEE. DOI: 10.1109/ciees55704.2022.9990861.

Rodriguez, G., Yannibelli, V., Rocha, F. G., Barbara, D., Azevedo, I. M., and Menezes, P. M. (2024). Understanding and addressing the allocation of microservices into containers: A review. *IETE Journal of Research*, 70(4):3887–3900. DOI: 10.1080/03772063.2023.2205864.

Rosa, G., Scalabrino, S., Bavota, G., and Oliveto, R. (2023). What quality aspects influence the adoption of docker images? *ACM Transactions on Software Engineering and Methodology*, 32(6):1–30. DOI: 10.1145/3603111.

Rosa, G., Zappone, F., Scalabrino, S., and Oliveto, R. (2024). Fixing dockerfile smells: an empirical study. *Empirical Software Engineering*, 29(5):108. DOI: 10.1007/s10664-024-10471-7.

Rosmaninho, R., Raposo, D., Rito, P., and Sargento, S. (2023). Time constraints on vehicular edge computing: A performance analysis. In *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pages 1–7. IEEE. DOI: 10.1109/noms56928.2023.10154306.

Roy, S., Dobson, R., Campbell, J., and Kalafatis, S. (2024). Efficient resource allocation for multi-robot collaboration via traffic-aware pod autoscaling. In *2024 7th Conference on Cloud and Internet of Things (CIoT)*, pages 1–8. IEEE. DOI: 10.1109/ciot63799.2024.10757045.

Ruiz Ródenas, J. M., Pastor-Galindo, J., and Gómez Mármol, F. (2024). A general and modular framework for dark web analysis. *Cluster Computing*, 27(4):4687–4703. DOI: 10.1007/s10586-023-04189-2.

Runeson, P. and Höst, M. (2009). Tutorial: Case studies in software engineering. *Lecture Notes in Business Information Processing*, 32 LNBIP:441–442. DOI: 10.1007/978-3-642-02152-7$_3$6.

Savitha, S., Sangana, C., Devendran, K., Pravin, L., Rajkumar, M., and Nirmal, C. (2023). Auto scaling infrastructure with monitoring tools using linux server on cloud. In *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 45–52. IEEE. DOI: 10.1109/iccmc56507.2023.10083635.

Saxena, A., Singh, S., Prakash, S., Yadav, N. L., Yang, T., Rathore, R. S., and Singh, S. (2024). Design and implementation of flutter based multi-platform docker controller app. In *2024 International Conference on Decision Aid Sciences and Applications (DASA)*, pages 1–6. IEEE. DOI: 10.1109/dasa63652.2024.10836516.

Saxena, V., Saxena, D., and Singh, U. P. (2022). Security enhancement using image verification method to secure docker containers. In *Proceedings of the 4th International Conference on Information Management & Machine Intelligence*, pages 1–5. DOI: 10.1145/3590837.3590879.

Schmidt, H., Rejiba, Z., Eidenbenz, R., and Förster, K.-T. (2023). Transparent fault tolerance for stateful applications in kubernetes with checkpoint/restore. In *2023 42nd International Symposium on Reliable Distributed Systems (SRDS)*, pages 129–139. IEEE. DOI: 10.1109/srds60354.2023.00022.

Sedov, D. and Lazarev, A. (2024). Enhancing it education through docker integration. In *2024 4th International Conference on Technology Enhanced Learning in Higher Education (TELE)*, pages 252–255. IEEE. DOI: 10.1109/tele62556.2024.10605628.

Segun-Falade, Osundare, Kedi, Okeleke, Ijomah, T., Abdul-Azeez, P., and Abdul-Azeez (2024). Assessing the transformative impact of cloud computing on software deployment and management. pages 2062–2082. DOI: 10.51594/csitrj.v5i8.1492.

Sepúlveda-Rodríguez, L. E., Chavarro-Porras, J. C., Sanabria-Ordoñez, J. A., Castro, H. E., and Matthews, J. (2022). A Survey of Virtualization Technologies: Towards a New Taxonomic Proposal. *Ingenieria e Investigacion*, 42(3). DOI: 10.15446/ing.investig.97363.

Shakya, S. and Tripathi, P. (2024). Using light weight container a mesh based dynamic allocation task scheduling algorithm for cloud with iot network. *International Journal of Information Technology*, 16(5):2847–2861. DOI: 10.1007/s41870-024-01772-7.

Shrestha, R. and Ray, A. (2024). Streamlining application deployment: A ci/cd pipeline for kubernetes. In *2024 IEEE International Conference on Cloud Engineering (IC2E)*, pages 253–255. IEEE. DOI: 10.1109/ic2e61754.2024.00038.

Silva, S. N., Goldbarg, M. A. d. S., Silva, L. M. d., and Fernandes, M. A. (2024). Application of fuzzy logic for horizontal scaling in kubernetes environments within the context of edge computing. *Future Internet*, 16(9):316. DOI: 10.3390/fi16090316.

Šimon, M., Huraj, L., and Búčik, N. (2023). A comparative analysis of high availability for linux container infrastructures. future internet, 15 (8).

Singh, K. U., Kumar, A., Kumar, G., Singh, T., Kothari, P., and Sheikh, A. (2024). Increasing productivity in software development through the use of docker technology. In *International Conference on Universal Threats in Expert Applications and Solutions*, pages 157–165. Springer. DOI: 10.1007/978-981-97-3810-6$_1$4.

Sobieraj, M. and Kotyński, D. (2024). Docker performance evaluation across operating systems. *Applied Sciences*, 14(15):6672. DOI: 10.3390/app14156672.

Soderi, M., Kamath, V., Morgan, J., and Breslin, J. G. (2022). Advanced analytics as a service in smart factories. In *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 000425–000430. IEEE. DOI:

10.1109/sami54271.2022.9780768.

Song, S., Suneja, S., Le, M. V., and Tak, B. (2023). On the value of sequence-based system call filtering for container security. In *2023 IEEE 16th International Conference on Cloud Computing (CLOUD)*, pages 296–307. IEEE. DOI: 10.1109/cloud60044.2023.00043.

Spahn, N., Hanke, N., Holz, T., Kruegel, C., and Vigna, G. (2023). Container orchestration honeypot: Observing attacks in the wild. In *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*, pages 381–396. DOI: 10.1145/3607199.3607205.

Stojanović, D., Jovanović, D., and Stojanović, N. (2024). Big medical data analytics using apache spark framework. In *2024 11th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN)*, pages 1–5. IEEE. DOI: 10.1109/IcETRAN62308.2024.10645083.

Thurimella, V., Raith, P., Enriquez, R. P. H., Da Silva, A. A., Rattihalli, G., Gavrilovska, A., and Milojicic, D. (2024). Serverless computing for dynamic hpc workflows. In *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 2096–2103. IEEE. DOI: 10.1109/scw63240.2024.00262.

Timonen, S., Sroor, M., Mohanani, R., and Mikkonen, T. (2023). Anomaly detection through container testing: A survey of company practices. In *International Conference on Product-Focused Software Process Improvement*, pages 363–378. Springer. DOI: 10.1007/978-3-031-49266-2$_2$5.

Ukene, D., Wimmer, H., and Kim, J. (2023). Evaluating the performance of containerized webservers against web servers on virtual machines using bombardment and siege. In *2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 144–152. IEEE. DOI: 10.1109/sera57763.2023.10197818.

Vaidya, R., Tembhurnikar, A., Mohite, C., Puri, S., Kulkarni, S., and Buchade, A. (2024). Blockchain-powered certificate authentication: Enhancing trust and transparency. In *2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)*, pages 1–5. IEEE. DOI: 10.1109/icbds61829.2024.10837062.

Vhatkar, K. N. and Bhole, G. (2022). A comprehensive survey on container resource allocation approaches in cloud computing: State-of-the-art and research challenges. *Web Intelligence*, 19(1):1–22. DOI: 10.3233/WEB-210474.

Voulgaris, K., Kiourtis, A., Karabetian, A., Karamolegkos, P., Poulakis, Y., Mavrogiorgou, A., and Kyriazis, D. (2022). A comparison of container systems for machine learning scenarios: Docker and podman. In *2022 2nd international conference on computers and automation (compauto)*, pages 114–118. IEEE. DOI: 10.1109/compauto55930.2022.00029.

VS, D. P., Sethuraman, S. C., and Khan, M. K. (2023). Container security: precaution levels, mitigation strategies, and research perspectives. *Computers & Security*, 135:103490. DOI: 10.1016/j.cose.2023.103490.

Wang, C., Liu, Z., Liu, J., Li, W., and Chen, J. (2024). Towards a container scheduling policy for alleviating total startup latency in serverless computing platform. In *Third international conference on algorithms, microchips, and network applications (AMNA 2024)*, volume 13171, pages 466–474. SPIE. DOI: 10.1117/12.3032003.

Wang, K., Wu, S., Suo, K., Liu, Y., Huang, H., Huang, Z., and Jin, H. (2023). Characterizing and optimizing kernel resource isolation for containers. *Future Generation Computer Systems*, 141:218–229. DOI: 10.1016/j.future.2022.11.018.

Wang, W. and Li, J. (2024). A feasibility study on containerization of traditional embedded systems. In *2024 4th International Conference on Electronic Information Engineering and Computer Communication (EIECC)*, pages 217–221. IEEE. DOI: 10.1109/eiecc64539.2024.10929091.

Waseem, M., Ahmad, A., Liang, P., Akbar, M. A., Khan, A. A., Ahmad, I., Setälä, M., and Mikkonen, T. (2025). Containerization in multi-cloud environment: roles, strategies, challenges, and solutions for effective implementation. *Journal of Systems and Software*, page 112558. DOI: 10.1016/j.jss.2025.112558.

Waseem, M., Ahmad, A., Liang, P., Akbar, M. A., Khan, A. A., Ahmad, I., Setälä, M., and Mikkonen, T. (2024). Containerization in multi-cloud environment: Roles, strategies, challenges, and solutions for effective implementation. *arXiv preprint*. Preprint accepted for publication in Journal of Systems and Software (2025). Submitted 1 Mar 2024; last revised 8 Jul 2025 (version v3).. DOI: 10.48550/arXiv.2403.12980.

Widodo, D., Kristalina, P., Hadi, M. Z. S., and Kurniawati, A. D. (2023). Performance evaluation of docker containers for disaster management dashboard web application. In *2023 International Electronics Symposium (IES)*, pages 551–556. IEEE. DOI: 10.1109/ies59143.2023.10242411.

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, EASE '14, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/2601248.2601268.

Wong, A. Y., Chekole, E. G., Ochoa, M., and Zhou, J. (2023). On the security of containers: Threat modeling, attack analysis, and mitigation strategies. *Computers & Security*, 128:103140. DOI: 10.1016/j.cose.2023.103140.

Wu, G., Li, X., and Li, C. (2024). A load balancing-based container migration mechanism in the smart grid cloud platform. In *2024 9th International Conference on Computer and Communication Systems (ICCCS)*, pages 1100–1104. IEEE. DOI: 10.1109/icccs61882.2024.10602810.

Wu, S., Tao, Z., Fan, H., Huang, Z., Zhang, X., Jin, H., Yu, C., and Cao, C. (2022). Container lifecycle-aware scheduling for serverless computing. *Software: Practice and Experience*, 52(2):337–352. DOI: 10.1002/spe.3016.

Xi, H., Hu, S., Zhang, H., Zhang, S., Weng, J., and Huang, J. (2024). A dynamic load balancing algorithm for stable pushing of docker images in nginx. In *2024 6th International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, pages 239–243. IEEE. DOI: 10.1109/mlbdbi63974.2024.10823883.

Xie, Y., Jin, M., Zou, Z., Xu, G., Feng, D., Liu, W., and Long, D. (2020). Real-time prediction of docker container resource load based on a hybrid model of arima and triple exponential smoothing. *IEEE Transactions on Cloud Computing*, 10(2):1386–1401. DOI: 10.1109/tcc.2020.2989631.

Xu, Y., Zhou, M., Gao, Q., Zhang, S., and Wu, Z. (2024). Swat4j: Generating system call allowlist for java container attack surface reduction. In *2024 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 929–939. IEEE. DOI: 10.1109/saner60148.2024.00101.

YAGHMAEI, F. (2003). Content validity and its estimation. *JOURNAL OF MEDICAL EDUCATION*. Available at:`https://brieflands.com/journals/jme/articles/105015`.

Yang, D. and Dai, W. (2022). A lightweight container design for microservice-based industrial edge applications. In *2022 IEEE 17th Conference on Industrial Electronics and Applications (ICIEA)*, pages 858–863. IEEE. DOI: 10.1109/iciea54703.2022.10006175.

Yang, N., Shen, W., Li, J., Liu, X., Guo, X., and Ma, J. (2023). Take over the whole cluster: Attacking kubernetes via excessive permissions of third-party applications. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 3048–3062. DOI: 10.1145/3576915.3623121.

Yang, Y., Shen, W., Xie, X., Lu, K., Wang, M., Zhou, T., Qin, C., Yu, W., and Ren, K. (2022). Making memory account accountable: Analyzing and detecting memory missing-account bugs for container platforms. In *Proceedings of the 38th Annual Computer Security Applications Conference*, pages 869–880. DOI: 10.1145/3564625.3564634.

Yaory, E. and Manuaba, I. B. K. (2024). A comparative study of complex query performance on mysql and oracle databases for the oil and gas industry. In *2024 International Conference on Information Management and Technology (ICIMTech)*, pages 1–6. IEEE. DOI: 10.1109/icimtech63123.2024.10780821.

Yarmilko, A., Naumenko, S., *et al.* (2023). Analysis of information security issues in balancing multiple independent containers on a single server. In *Proceedings of the 3rd International Workshop on Information Technologies: Theoretical and Applied Problems*, pages 450–461. Available at:`https://eprints.cdu.edu.ua/6022/`.

Ye, Y., Zhang, D., and Feng, G. (2024). A patch-based network model for container workload forecasting. In *Proceedings of the 2024 International Conference on Cloud Computing and Big Data*, pages 272–277. DOI: 10.1145/3695080.3695128.

Yin, J., Zhao, Y., and Wang, H. (2024). A static task allocation and scheduling algorithm for kubernetes cluster. In *2024 IEEE 7th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, pages 175–179. IEEE. DOI: 10.1109/iciscae62304.2024.10761792.

Yu, X., Zhang, X., Shi, Y., Wu, C., and Li, X. (2022). Design and implementation of vsto-based online compilation teaching system for c language. In *Proceedings of the 5th International Conference on Computer Science and Software Engineering*, pages 481–486. DOI: 10.1145/3569966.3570094.

Yuan, H. and Liao, S. (2024). A time series-based approach to elastic kubernetes scaling. *Electronics*, 13(2):285. DOI: 10.3390/electronics13020285.

Zehra, S., Syed, H. J., Samad, F., Faseeha, U., Ahmed, H., and Khan, M. K. (2024). Securing the shared kernel: Exploring kernel isolation and emerging challenges in modern cloud computing. *IEEE Access*. DOI: 10.1109/access.2024.3507215.

Zeng, Q., Kavousi, M., Luo, Y., Jin, L., and Chen, Y. (2023). Full-stack vulnerability analysis of the cloud-native platform. *Computers & Security*, 129:103173. DOI: 10.1016/j.cose.2023.103173.

Zhang, J., Yu, H., Fan, G., and Li, Z. (2022). Cold-start aware cloud-native service function chain caching in resource-constrained edge: A reinforcement learning approach. *Computer Communications*, 195:334–345. DOI: 10.1016/j.comcom.2022.09.004.

Zhang, Q., Ma, J., Zhang, X., and Liu, Y. (2024). Container security assessment and reinforcement technology integrating big data and intelligent algorithms. In *2024 3rd International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI)*, pages 622–626. IEEE. DOI: 10.1109/icdacai65086.2024.00119.

Zheng, T., Tang, R., Chen, X., and Shen, C. (2024). Kubefuzzer: Automating restful api vulnerability detection in kubernetes. *Computers, Materials & Continua*, 81(1). DOI: 10.32604/cmc.2024.055180.

Zhou, A. C., Huang, R., Ke, Z., Li, Y., Wang, Y., and Mao, R. (2024). Tackling cold start in serverless computing with multi-level container reuse. In *2024 IEEE international parallel and distributed processing symposium (IPDPS)*, pages 89–99. IEEE. DOI: 10.1109/ipdps57955.2024.00017.

Zhou, Y., Zhan, W., Li, Z., Han, T., Chen, T., and Gall, H. (2023). Drive: Dockerfile rule mining and violation detection. *ACM Transactions on Software Engineering and Methodology*, 33(2):1–23. DOI: 10.1145/3617173.

Zuppelli, M., Guarascio, M., Caviglione, L., and Liguori, A. (2024). No country for leaking containers: Detecting exfiltration of secrets through ai and syscalls. In *Proceedings of the 19th International Conference on Availability, Reliability and Security*, pages 1–8. DOI: 10.1145/3664476.3670884.

A Complete List of Selected Primary Studies

Table 12 provides the complete list of the 226 selected primary studies (SPS) included in this SMS.

ID	Ref	ID	Ref
SPS001	Pastor-Galindo *et al.*	SPS002	Moysiadis *et al.*
SPS003	Malviya and Dwivedi	SPS004	Šimon *et al.*
SPS005	Yaory and Manuaba	SPS006	Kamieniarz and Mazurczyk
SPS007	Voulgaris *et al.*	SPS008	Nakarmi *et al.*
SPS009	Chen *et al.*	SPS010	Betz *et al.*
SPS011	Xi *et al.*	SPS012	Li *et al.*
SPS013	Madi and Esteves-Verissimo	SPS014	Wang and Li
SPS015	Raj	SPS016	Modey *et al.*
SPS017	Yang and Dai	SPS018	Wu *et al.*
SPS019	Bracke *et al.*	SPS020	Abas *et al.*
SPS021	Fischer *et al.*	SPS022	Li *et al.*
SPS023	Deng *et al.*	SPS024	Yin *et al.*
SPS025	Malhotra *et al.*	SPS026	Yuan and Liao
SPS027	González-Abad *et al.*	SPS028	Ruiz Ródenas *et al.*
SPS029	Ebrahimpour *et al.*	SPS030	Ye *et al.*
SPS031	Liagkou *et al.*	SPS032	Baresi *et al.*
SPS033	Ghorbian and Ghobaei-Arani	SPS034	Aktolga *et al.*
SPS035	Joraviya *et al.*	SPS036	Nakakaze *et al.*
SPS037	Soderi *et al.*	SPS038	Qian
SPS039	Galantino *et al.*	SPS040	Aldiabat *et al.*
SPS041	Kumar and Kaur	SPS042	Aung *et al.*
SPS043	Dimova *et al.*	SPS044	Azuma *et al.*
SPS045	Ndigande *et al.*	SPS046	Husain *et al.*
SPS047	Yarmilko *et al.*	SPS048	de Oliveira Filho *et al.*
SPS049	Ajith *et al.*	SPS050	Timonen *et al.*
SPS051	Kotenko *et al.*	SPS052	Silva *et al.*
SPS053	Hettiarachchi *et al.*	SPS054	Fava *et al.*
SPS055	Savitha *et al.*	SPS056	Dogani *et al.*
SPS057	Purahong *et al.*	SPS058	Benzi *et al.*
SPS059	Chaurasia *et al.*	SPS060	Kaiser *et al.*
SPS061	Stojanović *et al.*	SPS062	Kanagachalam *et al.*
SPS063	Vaidya *et al.*	SPS064	Jolak *et al.*
SPS065	Hao *et al.*	SPS066	Blanco *et al.*
SPS067	Wang *et al.*	SPS068	Naydenov and Ruseva
SPS069	Ganne	SPS070	Raj *et al.*
SPS071	Rashid and Qasha	SPS072	Pavao *et al.*
SPS073	Zhang *et al.*	SPS074	Candemir and İncereis
SPS075	Choi *et al.*	SPS076	Pankowski and Powroźnik
SPS077	Moussa *et al.*	SPS078	Geng *et al.*
SPS079	Chen *et al.*	SPS080	Gao *et al.*
SPS081	Spahn *et al.*	SPS082	Zhang *et al.*
SPS083	Kaiser *et al.*	SPS084	Patra *et al.*
SPS085	Wu *et al.*	SPS086	VS *et al.*
SPS087	Waseem *et al.*	SPS088	Bentaleb *et al.*
SPS089	Malan	SPS090	Keller Tesser and Borin
SPS091	Kaur	SPS092	El Khairi *et al.*
SPS093	Choi *et al.*	SPS094	Alelyani *et al.*
SPS095	Joraviya *et al.*	SPS096	Nelson and Shoshitaishvili
SPS097	Zhou *et al.*	SPS098	Bentaleb *et al.*
SPS099	Kim *et al.*	SPS100	Saxena *et al.*
SPS101	Yu *et al.*	SPS102	Horchulhack *et al.*
SPS103	Chamoli and Mittal	SPS104	Sobieraj and Kotyński
SPS105	Patra *et al.*	SPS106	Gharaibeh *et al.*

Continues on the next page

Table 12 – continued

ID	Ref	ID	Ref
SPS107	Dipta *et al.*	SPS108	Gu *et al.*
SPS109	Jeon *et al.*	SPS110	Roy *et al.*
SPS111	Karumudi *et al.*	SPS112	Barbie *et al.*
SPS113	Ramanathan *et al.*	SPS114	Lee *et al.*
SPS115	Sedov and Lazarev	SPS116	Kostolny *et al.*
SPS117	Jang and Luo	SPS118	Flora and Antunes
SPS119	Ukene *et al.*	SPS120	Molnár *et al.*
SPS121	Dakić *et al.*	SPS122	Kaiser *et al.*
SPS123	Barletta *et al.*	SPS124	Rosa *et al.*
SPS125	Barros *et al.*	SPS126	Zeng *et al.*
SPS127	Gupta *et al.*	SPS128	Frasão *et al.*
SPS129	Gamess and Parajuli	SPS130	Alif and Munggaran
SPS131	Moric *et al.*	SPS132	Eroshkin *et al.*
SPS133	Singh *et al.*	SPS134	Kuity and Peddoju
SPS135	Narasimhulu *et al.*	SPS136	Entrialgo *et al.*
SPS137	Dogani *et al.*	SPS138	Lee *et al.*
SPS139	Ma *et al.*	SPS140	Kosińska *et al.*
SPS141	Zheng *et al.*	SPS142	Bellavista *et al.*
SPS143	Johansson *et al.*	SPS144	Carrión
SPS145	Carrión	SPS146	Botez *et al.*
SPS147	Haq *et al.*	SPS148	Dubey *et al.*
SPS149	Bannon	SPS150	Abbadini *et al.*
SPS151	Geetha *et al.*	SPS152	Fernalld *et al.*
SPS153	Mills *et al.*	SPS154	Han *et al.*
SPS155	Yang *et al.*	SPS156	Karmakar and Arri
SPS157	Mailewa *et al.*	SPS158	Barnawi *et al.*
SPS159	Pérez *et al.*	SPS160	Barletta *et al.*
SPS161	Zuppelli *et al.*	SPS162	Bhuiyan *et al.*
SPS163	Mondal *et al.*	SPS164	Mondal *et al.*
SPS165	Wong *et al.*	SPS166	Song *et al.*
SPS167	Bracke *et al.*	SPS168	Alamoush and Eichelberger
SPS169	Joshi *et al.*	SPS170	Kumar *et al.*
SPS171	Mthembu *et al.*	SPS172	Eng *et al.*
SPS173	Kurniawan *et al.*	SPS174	Melo *et al.*
SPS175	Widodo *et al.*	SPS176	Kithulwatta *et al.*
SPS177	Fu *et al.*	SPS178	Abdulah *et al.*
SPS179	Jeong *et al.*	SPS180	Gackstatter *et al.*
SPS181	Ersted Rasmussen *et al.*	SPS182	Xie *et al.*
SPS183	Lee *et al.*	SPS184	Karamzadeh and Shameli-Sendi
SPS185	Alyas *et al.*	SPS186	Mehran and Ulus
SPS187	Du *et al.*	SPS188	Xu *et al.*
SPS189	Al-Obaidi *et al.*	SPS190	Zehra *et al.*
SPS191	Haq *et al.*	SPS192	Saxena *et al.*
SPS193	Rajasekar *et al.*	SPS194	Thurimella *et al.*
SPS195	Al Qausar *et al.*	SPS196	Shrestha and Ray
SPS197	Agrawal and Singh	SPS198	Antonova *et al.*
SPS199	Burchart and Haake	SPS200	Mujkanovic *et al.*
SPS201	Hristev *et al.*	SPS202	Zhou *et al.*
SPS203	Yang *et al.*	SPS204	Jackson and Wurst
SPS205	Li *et al.*	SPS206	Ashari *et al.*
SPS207	Dobslaw *et al.*	SPS208	Rosmaninho *et al.*
SPS209	Álvarez *et al.*	SPS210	Wang *et al.*
SPS211	Amoiridis *et al.*	SPS212	Schmidt *et al.*
SPS213	Augustyn *et al.*	SPS214	Choi *et al.*
SPS215	Rodriguez *et al.*	SPS216	Kunekar *et al.*
SPS217	Shakya and Tripathi	SPS218	Arifiansyah *et al.*
SPS219	Kwon *et al.*	SPS220	Haresh *et al.*

Continues on the next page

Table 12 – continued

ID	Ref	ID	Ref
SPS221	Malhotra *et al*.	SPS222	Kjorveziroski and Filiposka
SPS223	Kjorveziroski and Filiposka	SPS224	Li *et al*.
SPS225	Rosa *et al*.	SPS226	Kim *et al*.

B Detailed Classification Tables

Table 13 presents the full classification of SPS studies by topic and year.

Table 13. Classification of SPS studies by topic and year

RQ	Topics	2022	2023	2024
Q1	Research	SPS002, SPS003, SPS007, SPS013, SPS017, SPS023, SPS039, SPS041, SPS044, SPS053, SPS059, SPS062, SPS064, SPS069, SPS070, SPS071, SPS073, SPS077, SPS079, SPS080, SPS083, SPS085, SPS088, SPS092, SPS098, SPS099, SPS105, SPS137, SPS143, SPS144, SPS145, SPS146, SPS149, SPS154, SPS155, SPS157, SPS176, SPS177, SPS180, SPS182, SPS187, SPS191, SPS192	SPS004, SPS012, SPS015, SPS027, SPS029, SPS034, SPS046, SPS047, SPS055, SPS056, SPS057, SPS060, SPS066, SPS067, SPS068, SPS075, SPS076, SPS081, SPS084, SPS086, SPS090, SPS093, SPS094, SPS097, SPS103, SPS117, SPS119, SPS126, SPS127, SPS134, SPS135, SPS138, SPS150, SPS153, SPS159, SPS165, SPS166, SPS167, SPS171, SPS173, SPS174, SPS175, SPS179, SPS183, SPS185, SPS189, SPS195, SPS200, SPS201, SPS203, SPS205, SPS208, SPS209, SPS212, SPS220, SPS221, SPS222, SPS223, SPS225, SPS226	SPS001, SPS005, SPS006, SPS008, SPS009, SPS010, SPS011, SPS014, SPS016, SPS018, SPS019, SPS021, SPS022, SPS024, SPS025, SPS026, SPS028, SPS030, SPS032, SPS033, SPS035, SPS036, SPS040, SPS043, SPS045, SPS048, SPS049, SPS050, SPS051, SPS052, SPS054, SPS061, SPS065, SPS074, SPS082, SPS087, SPS091, SPS095, SPS100, SPS102, SPS104, SPS106, SPS107, SPS108, SPS109, SPS110, SPS111, SPS113, SPS118, SPS121, SPS122, SPS123, SPS124, SPS125, SPS128, SPS129, SPS130, SPS131, SPS132, SPS133, SPS136, SPS140, SPS141, SPS142, SPS147, SPS148, SPS151, SPS156, SPS158, SPS160, SPS161, SPS162, SPS164, SPS168, SPS169, SPS170, SPS172, SPS178, SPS184, SPS186, SPS188, SPS190, SPS193, SPS194, SPS196, SPS197, SPS198, SPS202, SPS210, SPS213, SPS214, SPS215, SPS216, SPS217, SPS219, SPS224
	Education	SPS038, SPS058, SPS101, SPS146, SPS187, SPS204	SPS020, SPS072, SPS075, SPS116, SPS120, SPS152, SPS206, SPS207, SPS218	SPS042, SPS089, SPS096, SPS115, SPS124, SPS139, SPS151, SPS163, SPS198, SPS199
	Outreach	SPS002, SPS031, SPS037, SPS099	SPS078, SPS112, SPS208, SPS220	SPS010, SPS063, SPS114, SPS181, SPS211, SPS213

RQ	Topics	2022	2023	2024
Q2	Software development	SPS002, SPS037, SPS038, SPS044, SPS053, SPS058, SPS098, SPS101	SPS015, SPS078, SPS086, SPS120, SPS183, SPS195	SPS008, SPS010, SPS022, SPS028, SPS042, SPS043, SPS096, SPS100, SPS118, SPS133, SPS172, SPS215, SPS224
	Computational thinking	SPS187	SPS116	SPS042, SPS115, SPS198
	Parallel computing	SPS017	SPS020, SPS134, SPS223	
	Data analysis	SPS037, SPS071, SPS157	SPS183, SPS209	SPS001, SPS005, SPS028, SPS045, SPS061, SPS082, SPS129
	Artificial intelligence	SPS023, SPS037, SPS053, SPS059, SPS073, SPS077, SPS080, SPS149, SPS154, SPS177	SPS027, SPS072, SPS078, SPS183, SPS209	SPS011, SPS030, SPS040, SPS051, SPS082, SPS095, SPS142, SPS148, SPS161, SPS169, SPS170, SPS181
	Computer networks	SPS105, SPS187	SPS046, SPS094, SPS103, SPS159	SPS010, SPS019, SPS048, SPS106, SPS110, SPS113, SPS132, SPS139, SPS164, SPS198, SPS216, SPS219
	IT infrastructure	SPS003, SPS007, SPS017, SPS023, SPS031, SPS037, SPS038, SPS039, SPS062, SPS069, SPS070, SPS073, SPS077, SPS079, SPS083, SPS085, SPS088, SPS092, SPS099, SPS105, SPS137, SPS143, SPS144, SPS145, SPS146, SPS149, SPS154, SPS155, SPS176, SPS177, SPS180, SPS182, SPS187, SPS204	SPS004, SPS012, SPS020, SPS027, SPS029, SPS034, SPS046, SPS047, SPS055, SPS056, SPS057, SPS060, SPS066, SPS067, SPS068, SPS075, SPS076, SPS078, SPS081, SPS084, SPS090, SPS094, SPS103, SPS112, SPS117, SPS119, SPS126, SPS134, SPS135, SPS150, SPS152, SPS159, SPS167, SPS171, SPS173, SPS174, SPS175, SPS179, SPS183, SPS185, SPS189, SPS200, SPS201, SPS205, SPS206, SPS207, SPS208, SPS212, SPS218, SPS220, SPS222, SPS223, SPS225	SPS009, SPS011, SPS014, SPS018, SPS019, SPS021, SPS024, SPS025, SPS026, SPS030, SPS032, SPS033, SPS036, SPS048, SPS049, SPS051, SPS052, SPS054, SPS074, SPS082, SPS087, SPS089, SPS091, SPS095, SPS096, SPS100, SPS102, SPS104, SPS106, SPS107, SPS109, SPS110, SPS111, SPS115, SPS121, SPS122, SPS123, SPS124, SPS125, SPS129, SPS130, SPS131, SPS132, SPS136, SPS140, SPS148, SPS151, SPS156, SPS160, SPS163, SPS164, SPS168, SPS169, SPS170, SPS172, SPS178, SPS181, SPS184, SPS186, SPS188, SPS190, SPS196, SPS197, SPS198, SPS199, SPS210, SPS211, SPS213, SPS214, SPS215, SPS216, SPS217, SPS219, SPS224
	HPC	SPS017, SPS041, SPS062, SPS083, SPS098	SPS027, SPS090, SPS134, SPS200	SPS008, SPS014, SPS018, SPS114, SPS121, SPS129, SPS178, SPS194
	Blockchain			SPS063

RQ	Topics	2022	2023	2024
	Security	SPS013, SPS064, SPS069, SPS070, SPS079, SPS083, SPS092, SPS155, SPS157, SPS191, SPS192	SPS034, SPS047, SPS081, SPS086, SPS093, SPS094, SPS097, SPS119, SPS126, SPS127, SPS138, SPS150, SPS153, SPS165, SPS166, SPS175, SPS183, SPS189, SPS203, SPS221, SPS226	SPS001, SPS006, SPS009, SPS016, SPS022, SPS025, SPS035, SPS040, SPS043, SPS049, SPS050, SPS051, SPS065, SPS082, SPS108, SPS118, SPS125, SPS128, SPS129, SPS131, SPS141, SPS147, SPS156, SPS158, SPS160, SPS161, SPS162, SPS170, SPS188, SPS190, SPS193, SPS214, SPS219
	Cloud computing	SPS002, SPS003, SPS031, SPS069, SPS070, SPS071, SPS079, SPS080, SPS085, SPS099, SPS137, SPS143, SPS146, SPS149, SPS177	SPS012, SPS015, SPS029, SPS055, SPS056, SPS084, SPS126, SPS173, SPS179, SPS185, SPS222	SPS018, SPS019, SPS025, SPS026, SPS030, SPS032, SPS033, SPS043, SPS045, SPS087, SPS091, SPS109, SPS111, SPS136, SPS163, SPS193, SPS194, SPS197, SPS202, SPS210, SPS213, SPS214, SPS216, SPS217

C Quality Index Tables

Tables 14, 15, and 16 present the studies with the highest quality index scores, disaggregated by topic and year.

Table 14. Studies with the highest CVI index, classified by topics

RQ	Topics	2022	2023	2024
Q1	Research	SPS003, SPS007, SPS083, SPS145, SPS146	SPS068, SPS174	SPS032, SPS136, SPS151, SPS168
	Education	SPS038, SPS146	SPS152, SPS206	SPS089, SPS115, SPS151
Q2	Software Development	SPS038		
	Computational Thinking			SPS115
	Data Analysis	SPS037, SPS071, SPS157	SPS183, SPS209	SPS001, SPS005, SPS028, SPS045, SPS061, SPS082, SPS129
	IT Infrastructure	SPS003, SPS007, SPS038, SPS083, SPS145, SPS146	SPS068, SPS152, SPS174, SPS206	SPS032, SPS089, SPS115, SPS136, SPS151, SPS168
	HPC	SPS083		
	Security	SPS083		
	Cloud Computing	SPS003, SPS146		SPS032, SPS136

Table 15. Studies with the Highest SCI Index, Categorized by Topic

RQ	Topics	2022	2023	2024
Q1	Research	SPS003, SPS044, SPS064, SPS083, SPS092, SPS137, SPS143, SPS145, SPS157, SPS176, SPS187, SPS192	SPS027, SPS029, SPS126, SPS165, SPS173, SPS223	SPS028, SPS032, SPS033, SPS054, SPS140, SPS197, SPS215

RQ	Topics	2022	2023	2024
	Education	SPS187	SPS020, SPS072	
Q2	Software development	SPS044		SPS028, SPS215
	Computational thinking	SPS187		
	Parallel computing		SPS020, SPS223	
	Data analysis	SPS157		SPS028
	Artificial intelligence		SPS027, SPS072	
	Computer networks	SPS187		
	IT infrastructure	SPS003, SPS083, SPS092, SPS137, SPS143, SPS145, SPS176, SPS187	SPS020, SPS027, SPS029, SPS126, SPS173, SPS223	SPS032, SPS033, SPS054, SPS140, SPS197, SPS215
	HPC	SPS083	SPS027	
	Security	SPS064, SPS083, SPS092, SPS157, SPS192	SPS126, SPS165	
	Cloud computing	SPS003, SPS137, SPS143	SPS029, SPS126, SPS173	SPS032, SPS033, SPS197

Table 16. Studies with the Highest IRRQ Index, Classified by Topic

RQ	Topics	2022	2023	2024
Q1	Research	SPS002, SPS003, SPS007, SPS039, SPS044, SPS053, SPS059, SPS064, SPS070, SPS071, SPS073, SPS080, SPS083, SPS092, SPS137, SPS143, SPS145, SPS146, SPS155, SPS157, SPS176, SPS177, SPS187, SPS192	SPS027, SPS029, SPS055, SPS066, SPS067, SPS068, SPS081, SPS093, SPS094, SPS117, SPS126, SPS134, SPS153, SPS165, SPS167, SPS173, SPS174, SPS183, SPS195, SPS205, SPS209, SPS221, SPS223, SPS226	SPS005, SPS008, SPS010, SPS019, SPS021, SPS028, SPS030, SPS032, SPS033, SPS036, SPS045, SPS048, SPS054, SPS061, SPS082, SPS106, SPS107, SPS113, SPS129, SPS136, SPS140, SPS151, SPS168, SPS172, SPS178, SPS184, SPS197, SPS198, SPS214, SPS215, SPS216, SPS219
	Education	SPS038, SPS058, SPS101, SPS146, SPS187, SPS204	SPS020, SPS072, SPS116, SPS120, SPS152, SPS206, SPS207, SPS218	SPS089, SPS096, SPS115, SPS151, SPS163, SPS198, SPS199
	Outreach	SPS002, SPS031, SPS037	SPS078, SPS112	SPS010, SPS063, SPS114
Q2	Software development	SPS002, SPS037, SPS038, SPS044, SPS053, SPS058, SPS101	SPS078, SPS120, SPS183, SPS195	SPS008, SPS010, SPS028, SPS096, SPS172, SPS215
	Computational thinking	SPS187	SPS116	SPS115, SPS198
	Parallel computing		SPS020, SPS134, SPS223	
	Data analysis	SPS037, SPS071, SPS157	SPS183, SPS209	SPS005, SPS028, SPS045, SPS061, SPS082, SPS129
	Artificial Intelligence	SPS073	SPS209	SPS082
	Computer networks	SPS187	SPS094	SPS010, SPS019, SPS048, SPS106, SPS113, SPS198, SPS216, SPS219

RQ	Topics	2022	2023	2024
	IT infrastructure	SPS003, SPS007, SPS031, SPS037, SPS038, SPS039, SPS070, SPS073, SPS083, SPS092, SPS137, SPS143, SPS145, SPS146, SPS155, SPS176, SPS177, SPS187, SPS204	SPS020, SPS027, SPS029, SPS055, SPS066, SPS067, SPS068, SPS078, SPS081, SPS094, SPS112, SPS117, SPS126, SPS134, SPS152, SPS167, SPS173, SPS174, SPS183, SPS205, SPS206, SPS207, SPS218, SPS223	SPS019, SPS021, SPS030, SPS032, SPS033, SPS036, SPS048, SPS054, SPS082, SPS089, SPS096, SPS106, SPS107, SPS115, SPS129, SPS136, SPS140, SPS151, SPS163, SPS168, SPS172, SPS178, SPS184, SPS197, SPS198, SPS199, SPS214, SPS215, SPS216, SPS219
	HPC	SPS083	SPS027, SPS134	SPS008, SPS114, SPS129, SPS178
	Security	SPS064, SPS070, SPS083, SPS092, SPS155, SPS157, SPS192	SPS081, SPS093, SPS094, SPS126, SPS153, SPS165, SPS183, SPS221, SPS226	SPS082, SPS129, SPS214, SPS219
	Cloud computing	SPS002, SPS003, SPS031, SPS070, SPS071, SPS080, SPS137, SPS143, SPS146, SPS177	SPS029, SPS055, SPS126, SPS173	SPS019, SPS030, SPS032, SPS033, SPS045, SPS136, SPS163, SPS197, SPS214, SPS216

D Technology and Domain Classification Tables

Table 17 classifies SPS studies by CBV technology and academic dimension. Table 18 classifies SPS studies by IT domain and academic dimension.

Table 17. Classification of SPS studies by technology of **VBC** and academic dimension

Topics	Education	Research	Outreach
CRI-O		SPS068, SPS083	
Containerd		SPS066, SPS068, SPS083, SPS223	

Topics	Education	Research	Outreach
Docker	SPS020, SPS038, SPS042, SPS058, SPS072, SPS089, SPS096, SPS101, SPS115, SPS116, SPS120, SPS124, SPS152, SPS187, SPS198, SPS199, SPS204, SPS206, SPS207, SPS218	SPS002, SPS004, SPS005, SPS007, SPS008, SPS011, SPS017, SPS021, SPS030, SPS039, SPS040, SPS041, SPS043, SPS044, SPS045, SPS046, SPS048, SPS049, SPS051, SPS053, SPS054, SPS055, SPS059, SPS060, SPS061, SPS065, SPS066, SPS071, SPS074, SPS079, SPS080, SPS081, SPS083, SPS093, SPS097, SPS099, SPS100, SPS102, SPS103, SPS104, SPS105, SPS106, SPS107, SPS119, SPS122, SPS124, SPS126, SPS129, SPS133, SPS153, SPS155, SPS172, SPS173, SPS174, SPS176, SPS177, SPS180, SPS182, SPS187, SPS188, SPS191, SPS192, SPS197, SPS198, SPS205, SPS209, SPS216, SPS219, SPS220, SPS221, SPS225, SPS226	SPS002, SPS037, SPS063, SPS078, SPS099, SPS112, SPS114, SPS220
Firecracker		SPS107, SPS205	
Google gVisor		SPS107, SPS184, SPS205	
Hyper-V containers		SPS068	
Kata Containers		SPS184, SPS205, SPS224	
LXC		SPS066, SPS068, SPS083, SPS157	
LXD		SPS068, SPS083	
OpenVZ		SPS083	
Podman		SPS007, SPS046, SPS060, SPS068, SPS083, SPS129, SPS174	
Rkt		SPS068, SPS083	
Singularity		SPS041, SPS060, SPS068	
Udocker		SPS027, SPS068	

Table 18. Classification of SPS studies by IT domain and academic dimension

Topics	Education	Research	Outreach
Data analysis		SPS001, SPS005, SPS028, SPS045, SPS061, SPS071, SPS082, SPS129, SPS157, SPS183, SPS209	SPS037
Blockchain			SPS063

Topics	Education	Research	Outreach
Cloud computing	SPS146, SPS163	SPS002, SPS003, SPS012, SPS015, SPS018, SPS019, SPS025, SPS026, SPS029, SPS030, SPS032, SPS033, SPS043, SPS045, SPS055, SPS056, SPS069, SPS070, SPS071, SPS079, SPS080, SPS084, SPS085, SPS087, SPS091, SPS099, SPS109, SPS111, SPS126, SPS136, SPS137, SPS143, SPS146, SPS149, SPS173, SPS177, SPS179, SPS185, SPS193, SPS194, SPS197, SPS202, SPS210, SPS213, SPS214, SPS216, SPS217, SPS222	SPS002, SPS031, SPS099, SPS213
Parallel computing	SPS020	SPS017, SPS134, SPS223	
Software development	SPS038, SPS042, SPS058, SPS096, SPS101, SPS120	SPS002, SPS008, SPS010, SPS015, SPS022, SPS028, SPS043, SPS044, SPS053, SPS086, SPS098, SPS100, SPS118, SPS133, SPS172, SPS183, SPS195, SPS215, SPS224	SPS002, SPS010, SPS037, SPS078
HPC		SPS008, SPS014, SPS017, SPS018, SPS027, SPS041, SPS062, SPS083, SPS090, SPS098, SPS121, SPS129, SPS134, SPS178, SPS194, SPS200	SPS114
Artificial intelligence	SPS072	SPS011, SPS023, SPS027, SPS030, SPS040, SPS051, SPS053, SPS059, SPS073, SPS077, SPS080, SPS082, SPS095, SPS142, SPS148, SPS149, SPS154, SPS161, SPS169, SPS170, SPS177, SPS183, SPS209	SPS037, SPS078, SPS181
Computational thinking	SPS042, SPS115, SPS116, SPS187, SPS198	SPS187, SPS198	
Computer networks	SPS139, SPS187, SPS198	SPS010, SPS019, SPS046, SPS048, SPS094, SPS103, SPS105, SPS106, SPS110, SPS113, SPS132, SPS159, SPS164, SPS187, SPS198, SPS216, SPS219	SPS010
Security		SPS010, SPS019, SPS046, SPS048, SPS094, SPS103, SPS105, SPS106, SPS110, SPS113, SPS132, SPS159, SPS164, SPS187, SPS198, SPS216, SPS219	

Topics	Education	Research	Outreach
IT infrastructure	SPS020, SPS038, SPS075, SPS089, SPS096, SPS115, SPS124, SPS146, SPS151, SPS152, SPS163, SPS187, SPS198, SPS199, SPS204, SPS206, SPS207, SPS218	SPS003, SPS004, SPS007, SPS009, SPS011, SPS012, SPS014, SPS017, SPS018, SPS019, SPS021, SPS023, SPS024, SPS025, SPS026, SPS027, SPS029, SPS030, SPS032, SPS033, SPS034, SPS036, SPS039, SPS046, SPS047, SPS048, SPS049, SPS051, SPS052, SPS054, SPS055, SPS056, SPS057, SPS060, SPS062, SPS066, SPS067, SPS068, SPS069, SPS070, SPS073, SPS074, SPS075, SPS076, SPS077, SPS079, SPS081, SPS082, SPS083, SPS084, SPS085, SPS087, SPS088, SPS090, SPS091, SPS092, SPS094, SPS095, SPS099, SPS100, SPS102, SPS103, SPS104, SPS105, SPS106, SPS107, SPS109, SPS110, SPS111, SPS117, SPS119, SPS121, SPS122, SPS123, SPS124, SPS125, SPS126, SPS129, SPS130, SPS131, SPS132, SPS134, SPS135, SPS136, SPS137, SPS140, SPS143, SPS144, SPS145, SPS146, SPS148, SPS149, SPS150 SPS151, SPS154, SPS155, SPS156, SPS159, SPS160, SPS164, SPS167, SPS168, SPS169, SPS170, SPS171, SPS172, SPS173, SPS174, SPS175, SPS176, SPS177, SPS178, SPS179, SPS180, SPS182, SPS183, SPS184, SPS185, SPS186, SPS187, SPS188, SPS189, SPS190, SPS196, SPS197, SPS198, SPS200, SPS201, SPS205, SPS208, SPS210, SPS212, SPS213, SPS214, SPS215, SPS216, SPS217, SPS219, SPS220, SPS222, SPS223, SPS224, SPS225	SPS031, SPS037, SPS078, SPS099, SPS112, SPS181, SPS208, SPS211, SPS213, SPS220