IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 15 March 2024, accepted 29 March 2024, date of publication 15 April 2024, date of current version 19 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3389064

TOPICAL REVIEW

Energy Consumption in Microservices Architectures: A Systematic Literature Review

GABRIEL ARAÚJO[1], VANDIRLEYA BARBOSA[1], LUIZ NELSON LIMA[1], ARTHUR SABINO[1], CARLOS BRITO[1], IURE FÉ[1], PAULO REGO[2], EUNMI CHOI[3], DUGKI MIN[4], TUAN ANH NGUYEN[5], (Member, IEEE), AND FRANCISCO AIRTON SILVA[1]

[1]PASID Laboratory, Federal University of Piauí (UFPI), Picos, Piauí 64607-670, Brazil
[2]Federal University of Ceará, Fortaleza 60020-181, Brazil
[3]School of Software, College of Computer Science, Kookmin University, Seoul 02707, South Korea
[4]Department of Computer Science and Engineering, College of Engineering, Konkuk University, Seoul 05029, South Korea
[5]Konkuk Aerospace Design-Airworthiness Institute (KADA), Konkuk University, Seoul 05029, South Korea

Corresponding authors: Francisco Airton Silva (faps@ufpi.edu.br), Dugki Min (dkmin@konkuk.ac.kr), and Tuan Anh Nguyen (anhnt2407@konkuk.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1A6A1A03046811 and Grant 2021R1A2C2094943.

ABSTRACT Cloud computing emerges as a paradigm that facilitates on-demand access to technological resources through the mechanism of service virtualization. This virtualization enables the partitioning of hardware resources among applications that are organized into distinct independent modules. The concept of microservice architecture takes advantage of virtualization capabilities to embrace a software architecture strategy focused on the development of applications as assemblies of several interdependent but loosely coupled modules. Nonetheless, the adoption of microservices architecture is accompanied by substantial energy demands to meet the desired standards of performance and availability. Existing research within the domain of microservices has explored various topics pertinent to energy consumption, including elasticity, reliability, performance, and availability. Yet, the diversity of challenges and solutions presents a complex landscape for identifying prevailing research trends and unaddressed gaps in the context of microservices. This study aims to methodically discern, evaluate, and juxtapose the existing research trends and voids concerning energy consumption within microservices. It elucidates a systematic review on the subject of energy consumption in microservices architectures, offering a compilation of references to facilitate more directed future investigations. The initial selection encompassed 3625 articles, which were subsequently narrowed down through three stages of refinement, resulting in 37 articles chosen for an exhaustive review. These selected studies were cataloged and analyzed based on various criteria, including metrics, evaluation methodologies, and architectural typologies, thus uncovering research gaps and emerging trends related to energy consumption in microservice architectures. Furthermore, this inquiry delineates significant research challenges and prospective directions, structured around the key metrics that underpin the reviewed studies: performance, elasticity, scalability, reliability, sustainability, and availability.

INDEX TERMS Microservices, systematic mapping, energy consumption, cloud computing, container.

I. INTRODUCTION

Cloud computing represents a paradigm that facilitates instantaneous network access to a collective pool of configurable computing resources, such as networks, servers, storage, applications, and services. These resources can be swiftly provisioned and relinquished with minimal management effort or service provider interaction. The proliferation of cloud computing has led to an increasing presence of virtualization in daily life, emphasizing cost

The associate editor coordinating the review of this manuscript and approving it for publication was Libo Huang.

reduction through consolidation, enhancement of system efficiency, and performance improvement. Rapid evolution has placed cloud computing at the forefront of contemporary technological trends, as evidenced by significant literature contributions [1], [2], [3], [4], [5], [6], [7]. Cloud computing's inception aimed at democratizing computing resource access, enabling both enterprises and individuals to avail high-caliber IT infrastructures without substantial hardware and software investments. Leading providers, including Amazon, Google, Salesforce, IBM, Microsoft, and Sun Microsystems, have expanded their data centers globally to improve application hosting for cloud computing, improving redundancy and reliability against site failures [8]. With advances in cloud computing, it now offers sophisticated virtualization, automation, and data management capabilities, finding applications across various sectors, from e-Commerce to scientific exploration.

Gartner Inc.'s latest projections indicate a 20.7% increase in global end-user spending on public cloud services, reaching $591.8 billion in 2023, up from $490.3 billion in 2022, outpacing the anticipated 18.8% growth in 2022 [9]. The microservices software architecture, notable for its reliance on virtualization and modularization, employs virtualization to enhance system infrastructure efficiency and modularization to segment systems into simpler, independent units. This architecture endorses the development of flexible, scalable, and maintainable systems [10], [11]. Microservices are defined as small, independent processes that fully implement specific functionalities, operating autonomously without external service dependencies. On the contrary, monolithic architectures contribute to system inflexibility by tightly coupling functions, complicating feature modifications, and inefficiently scaling unneeded features. Unlike monolithic systems where functionalities are interdependent, microservices architecture offers scalability and responsibility advantages, albeit with challenges in refactoring and distributed communications. The architecture's scalability permits targeted adjustments to individual microservices, thus enhancing organizational agility and delivery speed.

The advent of microservices architectures as an effective and adaptable method for developing and managing cloud applications allows for the distributed deployment across various cloud providers, presenting a unified service deployment illusion. This adaptability has spurred widespread adoption among major corporations [12], [13], with notable examples like Netflix, Amazon, and eBay leveraging the ''divide and conquer'' strategy to deconstruct monolithic applications into manageable, function-specific units. Despite the numerous benefits, microservice architectures also face challenges, particularly in energy efficiency and sustainability, which have significant environmental and operational cost implications [12], [13]. Addressing these challenges is crucial for exploiting microservices' full potential while mitigating their limitations.

Recent studies have engaged in performance evaluation of microservice architectures using simulation tools such as CloudSim [14], [15], explored energy management through elasticity metrics [16], [17], and assessed reliability through modeling approaches [18], [19]. Investigations reveal that microservices can incur substantial energy costs to maintain system availability, scalability, performance, reliability, and sustainability [18], [20], [21], [22], [23], [24], highlighting the importance of energy-efficient architectural strategies.

This paper undertakes a systematic review of energy consumption within microservices architectures, initiating with 3625 articles and narrowing down through successive refinement stages to 37 for detailed analysis. These articles were evaluated and categorized according to context, evaluation methodology, and architectural type, uncovering prevailing research gaps and trends in the energy consumption of microservices.

Subsequent sections are organized as follows. Section II delves into related works. Section III describes the methodology used. Section IV presents classification-based findings. Section V discusses a categorization derived from intersecting classifications. Section VI outlines identified challenges and future directions post-review. Lastly, Section VII concludes with recommendations for future research in this area.

II. RELATED WORKS

This section reviews pertinent literature within contexts similar to the focus of this study.

Vural et al. [25] engage in a systematic mapping study on microservices to discern contemporary trends, noting a gap in addressing energy efficiency challenges in microservices-based systems design, development, and maintenance, indicating a broader understanding gap on its implications for system performance and energy efficiency. Ghofrani and Lübke [26] detail an empirical study assessing the state of current practice and identifying microservices architecture challenges.

Viggiato et al. [27] share a survey with 122 professionals involved in microservices projects, exploring industry application of architecture and matching practitioners' perceptions with literature-reported advantages and challenges. Soldani et al. [28] systematically analyze unpublished industry (grey) literature on microservices, identifying technical and operational gains and objections.

Yu et al. [29] survey security risks in cloud-based microservices applications, emphasizing the architecture's inherent security issues due to frequent service communications and proposing solutions for service communication security. Di Francesco et al. [30] aim to classify and evaluate the state-of-the-art microservice architecture, noting several unexplored aspects.

Waseem et al. [31] conduct a systematic literature analysis on DevOps microservices architecture application, aiming to categorize and assess the state of the art. Hannousse and Yahiouche [32] guide developers on recognized threats in microservices, providing insights into detection, mitigation, and prevention strategies. Jawaddi et al. [33] review the literature on microservices, verification and automatic scaling, highlighting recent advancements between 2017 and 2022.

Table 1 categorizes related works based on context, criteria and study type, identifying systematic mapping or survey studies as a common classification criterion, highlighting the complexity and challenges of integrating services, managing distributed data and maintaining multiple services as key issues in microservice adoption.

In their scholarly contribution, Viggiato et al. [27] conduct an investigative survey that delves into the practical implementation and adoption nuances of microservices architectures. The authors elucidate the core motivations propelling the adoption of microservices, delineate the predominant challenges encountered during their deployment, and outline the strategic measures undertaken to mitigate these obstacles. Concurrently, Yu et al. [29] embark on a survey focused on the security intricacies inherent to the communication between fog application services, underpinned by microservices frameworks.

The research undertaken by Vural et al. [25] engages in a systematic mapping study to scrutinize various dimensions of microservices adoption. This study offers insights into the fundamental reasons driving the adoption of microservices, elucidates the attributes and advantages linked with microservices architectures, and explores the pivotal tools and technologies integral to their implementation. Similarly, Soldani et al. [28] present a systematic mapping study aimed at illuminating the principal challenges associated with the adoption of microservices, including the management of multiple service instances, the establishment of a robust communication framework, and the preservation of service cohesion. In another vein, Di Francesco et al. [30] conduct a systematic mapping study to probe into the research trajectories within the microservices domain, dissecting the evolution of research, the primary areas of inquiry, and the methodologies and techniques prevalently employed in microservices research.

Waseem et al. [31] offer a systematic mapping study that examines the integration of microservices architecture within DevOps practices, addressing both the challenges and benefits of this amalgamation, alongside the principal tools and techniques for its execution. The scholarly work of Hannousse and Yahiouche [32] unfolds as a systematic mapping study on the security aspects of microservices and architectures predicated on microservices. Jawaddi et al. [33] spotlight the leading approaches towards self-scaling microservices, along with the instruments and formal verification techniques deployed to ascertain their efficacy.

These related works are also methodically categorized based on the **context**, which underscores the central thematic focus of each study. The works by Vural et al. [25] and Di Francesco et al. [30] are primarily centered on microservices architectures, with an aim to discern trends, research domains, and the adoption potential of this architectural paradigm. In contrast, the investigations by Ghofrani and Lübke [26] and Viggiato et al. [27] delve into the empirical examination of microservices in practical scenarios, striving to comprehend the implementation challenges and glean insights into the operational facets of this architectural approach. Our research distinguishes itself by its singular focus on energy consumption within microservices architectures, exploring a distinctive angle that differentiates it from the extant body of work in the field.

The **criteria** column delineates the evaluative standards applied in the related works. These standards pertain to the overarching classification of findings across the studies. Key criteria encompass context, security, challenges, metrics, methods, and tools. The studies by Vural et al. [25], Viggiato et al. [27], Soldani et al. [28], and Di Francesco et al. [30] address the contextual backdrop of microservices utilization. Meanwhile, Yu et al. [29] and Hannousse and Yahiouche [32] scrutinize security considerations within microservices architecture. Challenges specific to microservices are examined by Ghofrani and Lübke [26] and Jawaddi et al. [33], with the latter also shedding light on the metrics prevalent in this sphere. Waseem et al. [31] describe the methodologies embraced in their study, whereas Vural et al. [25], Waseem et al. [31], and Hannousse et al. [32] elucidate the tools engaged in their respective inquiries. This work carves a niche by probing into the cloud architectures that underpin microservices usage.

III. METHODOLOGY

The methodology for the systematic review employed in this investigation draws inspiration from [32]. As depicted in Figure 1, the research methodology encompasses several stages: defining the scope, conducting the initial search, applying filtering criteria, classifying the findings, and extracting pertinent information. Within the scope definition phase, guiding questions were formulated to refine the search outcomes, thereby ensuring a focused investigation into the targeted subject matter. The culmination of these efforts is presented in the results section, showcasing the insights gleaned from this systematic review.

A. SCOPE DEFINITION

The initial phase of this research is devoted to defining the scope, which is established through a set of guiding questions. These questions delineate the boundaries of the study's domain, effectively narrowing down the research area to be explored. In preparation for crafting these questions, there was a collective agreement on focusing on pertinent topics that would steer the investigation towards "energy consumption in microservices." As a result, this study introduces three pivotal questions to refine the scope of the research:

TABLE 1. Related works.

Works	Year	Context	Criteria	Type of Study
[25]	2017	Microservices	Context/practical motivations/tools	Systematic mapping
[26]	2018	State of practice and collect challenges in microservices	Challenges/Solutions	Survey
[27]	2018	Microservices in Practice	Context	Survey
[28]	2018	Industrial application of microservices	Context	Systematic Review
[29]	2019	Services communication of Microservices	Security problems	Survey
[30]	2019	Architecting with Microservices	Context/problems	Systematic mapping
[31]	2020	Microservices Architecture in DevOps	Methods / Design Patterns / Quality Attribute / Tools	Systematic mapping
[32]	2021	Security in microservice architectures	Security threats/techniques and tools	Systematic mapping
[33]	2022	Microservices autoscaling with formal verification perspective	Autoscaling techniques/metrics/challenges	Systematic Review
Our work	2023	Energy consumption in Microservices	Context/Architectures/Evaluated methods/Metrics	Systematic Mapping

FIGURE 1. Flowchart of the methodology applied in the research.

- **Q1**: What are the various objectives associated with energy consumption in microservices? This inquiry seeks to uncover the underlying aims of research endeavors concerning energy consumption within microservices environments.
- **Q2**: Which architectures are involved in energy consumption concerns within microservices? This question seeks to pinpoint the architectural frameworks that research has identified as relevant to address energy consumption issues in microservices.
- **Q3**: What evaluation metrics are applied to assess energy consumption in microservices? The goal here is to examine the range of metrics used by researchers to gauge energy consumption in microservices setups.

energy AND (microservices OR container)

FIGURE 2. Used search string.

TABLE 2. Direct exclusion criteria.

Direct exclusion criteria
Work is not in English.
Work with less than six pages.
Non-scientific papers.
Being a Survey/Mapping/Review.
Work is not about power in microservices.

B. SEARCH

The subsequent phase focuses on the search process. During this stage, a specific search term was used in five distinct search engines to gather relevant papers. The selection of keywords was meticulously tailored to the theme "Energy in microservices," leading to the generation of a search string that prioritized "energy" as the essential term, while "microservices" and "container" were considered as secondary terms. Nevertheless, either "microservices" or "container" needed to be paired with "energy" in the search query.

The search string was adjusted to align with the indexing conventions of various databases, including IEEE Xplore, Elsevier, Springer, ACM, and Science Direct. Following the initial search and the accumulation of papers, a filtration process was applied as depicted in Figure 3. It is pertinent to note that the search string was executed through the advanced search functionalities offered by the search engines, targeting the occurrences of the search terms within both titles and abstracts of the publications. Figure 2 provides a visual representation of the constructed search string used in this investigation.

C. FILTERING

The third phase of the study encompasses the Filtering process. This stage incorporates a systematic approach to exclusion, leveraging predefined criteria outlined in Table 2 for direct removal of articles. The methodology for the inclusion of the article involved evaluating each article against these exclusion criteria. Papers that did not align with any of the stipulated reasons for exclusion were automatically advanced to subsequent phases of the filtering process. Figure 3 delineates the flowchart employed for executing the exclusion criteria, providing a visual representation of the entire filtering sequence enacted on the initial pool of 3625 papers. This flowchart elucidates the progression of papers through the filtering process, highlighting the number of papers retained or eliminated at each juncture, along with the specific reasons for their exclusion.

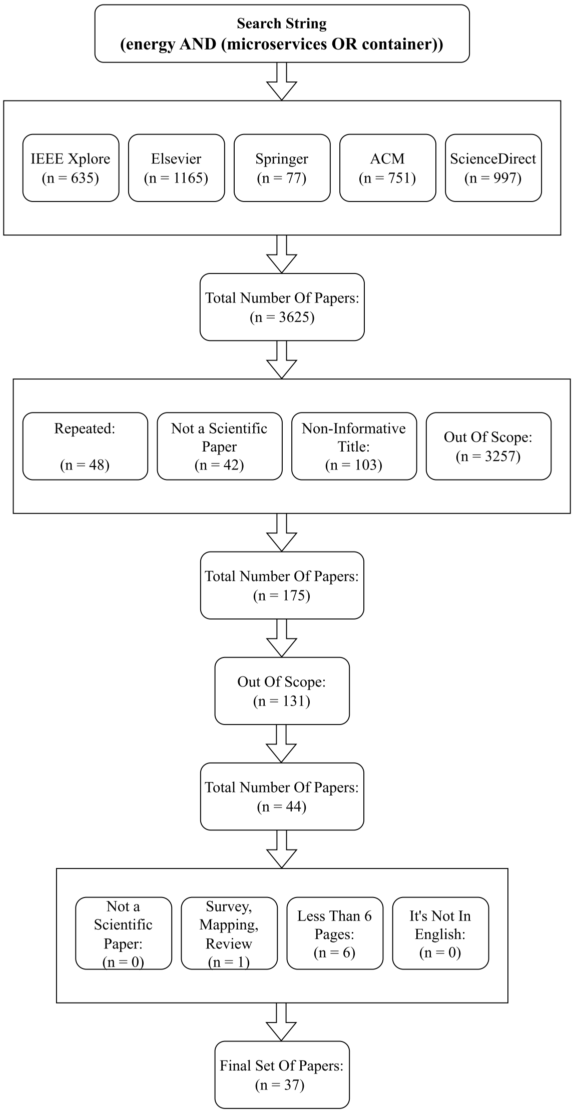


FIGURE 3. Flowchart that demonstrates the stages of the filtering process.

During the initial stage of filtering, an examination of the titles led to the exclusion of papers based on predetermined criteria such as non-relevance of title, topics falling outside the study's scope, duplications, and non-research paper formats, resulting in the elimination of 3436 papers. Subsequently, the abstracts of the remaining papers were scrutinized to assess their relevance to the research scope, leading to the further exclusion of 139 papers that did not align with the research objectives. The third stage entailed a comprehensive review of the remaining 50 articles in their entirety, applying the direct exclusion criteria outlined in Table 2, which culminated in the selection of 37 papers for in-depth analysis. The ensuing phase involved the categorization of these papers based on various factors such as Context, Macro Metrics, Evaluation Method, Type of Architecture, Orchestrator, and Deployment Model pertinent to each study.

TABLE 3. Categorization.

General Evaluation	Classification
Metrics	Performance Elasticity Scalability Reliability Sustainability Availability
Evaluation Method	Simulation Measurement Modeling
Type of Architecture	Cloud Computing Fog Computing Cloud Computing and Fog Computing
Orchestrator	Kubernetes Docker Swarm Uninformed
Deployment Model	Private Public Private and Public Private and Public and Hybrid Uninformed

D. CLASSIFICATION

The Classification phase, constituting the fourth step of the methodology, was aimed at facilitating a more nuanced extraction of data and pinpointing areas lacking in research. This stage of the classification of the work was divided into two distinct segments. The initial segment involved delineating the classification criteria, followed by the actual categorization of the articles based on these established criteria. Five primary categories were defined for this purpose: Context, Macro Metrics, Evaluation Method, Type of Architecture, Orchestrator, and Deployment Model. The subsequent segment entailed devising specific classifications within these categories. These classifications, detailed in Table 3, were derived from an analysis of the selected papers and the preliminary frameworks proposed by the researchers.

E. INFORMATION EXTRACTION

The Information Extraction process represents the fifth step in this research methodology. For this study, the extraction of pertinent information from the selected articles involved a detailed review and analysis of each work. This comprehensive evaluation aimed to gather critical data points, including the metrics applied, evaluation methodologies employed, architectural frameworks, tools, orchestrators, and deployment models utilized within each study. Furthermore, the papers were systematically classified to elucidate the main objectives and enhance understanding of the scope of the research.

IV. GENERAL CLASSIFICATION RESULTS

This section delineates the results derived from the classification process as previously outlined.

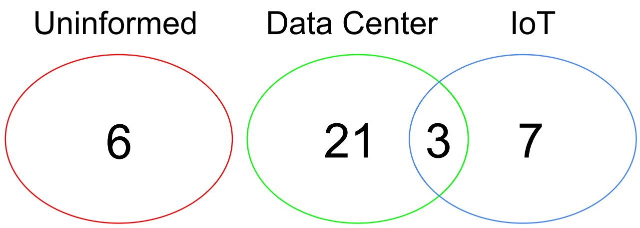


FIGURE 4. Venn diagram with the classification of mapped contexts.

A. CONTEXT

Presented in this subsection is a Venn diagram (refer to Figure 4) that encapsulates the classification of contexts where the architecture of microservices and the use of containers are leveraged for energy consumption optimization. This diagram offers a panoramic view of the varied contexts in which these technologies have been implemented, underscoring the need for more energy-efficient solutions. The critical examination of the specific contexts employing microservices and containers is essential, given that each scenario presents distinct characteristics and demands. The findings reveal that a significant majority (83.8%) of the studies explicitly delineate the context of use, while the rest (16. 2%) delineate the context primarily in the context of proposing strategies for the enhancement of energy efficiency in generic microservice frameworks. From the analysis, two predominant contexts have been identified: Data Centers and the Internet of Things (IoT).

In the study by Gholipoura et al. [14], a novel approach for the joint management of Virtual Machine (VM) and container resources in cloud data centers is proposed, demonstrating significant reductions in energy consumption. The research conducted by Silva de Souza et al. [34] underscores the importance of optimizing software operations to mitigate the financial and environmental impacts of data centers, emphasizing the increased complexity due to the expanded configuration space and greater heterogeneity of contemporary data center and cloud architectures. Ferreira et al. [35] introduce a middleware focused on energy management through edge computing for smart homes, while the study by Raza et al. [36] seeks to establish guidelines and identify critical factors for deploying containerized services on resource-constrained IoT devices.

1) DATA CENTER

Data Centers, composed of servers, storage units, networking devices, power distribution, and cooling systems [37], serve as pivotal infrastructure for organizations to store vast amounts of data and provide data processing services. Microservices-based architectures offer several benefits for data centers, including simplified management of multiple applications, scalable application infrastructure, enhanced availability, and increased flexibility. Khairy et al. [21] introduce the SIMR (Single Instruction Multiple Request Processing) technique, which processes multiple requests in a single cycle, thereby reducing processor instructions and network traffic, and significantly cutting down on data center energy consumption. Silva de Souza et al. [34] discuss the creation of Containergy, a tool for measuring the power and performance of containerized workloads in data centers.

Lin et al. [17] propose a method for the efficient allocation and deployment of containers in data centers to lower energy consumption. Kaur et al. [38] suggest the EnLoB (Energy and Load Balancing-Driven Container Placement Strategy) for energy-efficient container placement in data centers, aiming to balance load and enhance server performance. Patel et al. [39] introduce an energy-efficient genetic algorithm for container consolidation in cloud systems to minimize energy use in container-based virtualization data centers. Studies by Bouaouda et al. [23] and Khan et al. [15] explore strategies for reducing energy consumption in data centers without compromising on performance.

2) IOT

The Internet of Things (IoT) represents a transformative technology that changes our interaction with and control over the physical world, consisting of interconnected intelligent devices that gather and exchange data [40]. Microservices in IoT settings can facilitate the monitoring and management of energy consumption, promoting energy-efficient decision-making. Lennick et al. [41] detail a microservices-based architecture for assessing performance and power consumption of Linux distributions in Docker containers on IoT devices.

Jawar et al. [20] discuss the role of microservices in energy-data management infrastructures supported by IoT, where data from various energy generation and consumption sources in buildings is abundant. They advocate for the use of data processing microservice modelling to enhance data availability and offer microservices for energy efficiency management.

B. MACRO METRICS

Given the variety of relevant metrics utilized across the studies, they were categorized into two groups: Macro Metrics and Micro Metrics, where Macro Metrics represent broader categories of general metrics and Micro Metrics are specific metrics within a Macro Metric category. The increasing adoption of microservices, attributed to their flexibility, necessitates the monitoring of quality and performance through established metrics to assess service behavior, identify issues, and plan service improvements. Figure 5 illustrates the Macro Metrics identified in the analyzed papers, revealing five main categories: performance, accounted for in 59.5% of the studies; elasticity, covered in 29.7%; reliability, mentioned in 5.4%; sustainability, observed in 2.7%; and availability, also in 2.7% of the studies.

Figure 6 illustrates a taxonomy of metrics that encompasses both Macro and Micro Metrics. The architecture of microservices and containers is widely acknowledged for its

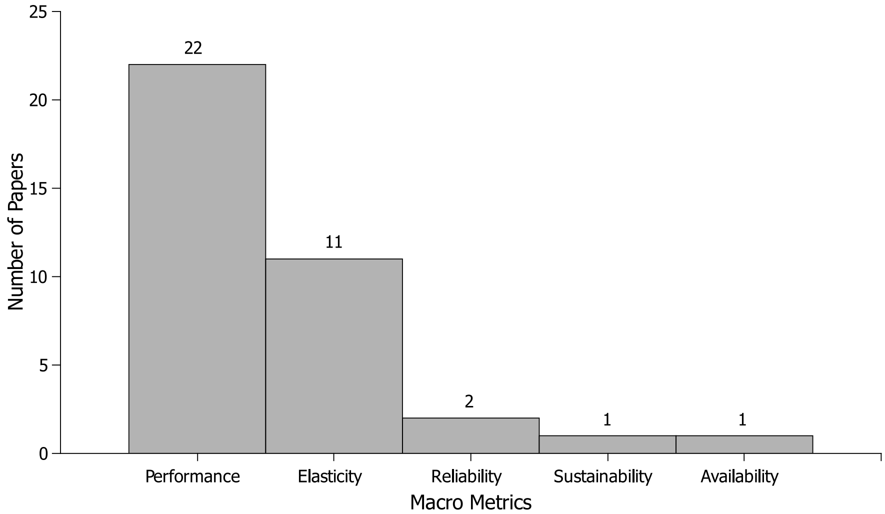


FIGURE 5. **Classification of metrics.**

potential to enhance the performance of software systems. However, gauging the efficiency of these systems, particularly concerning energy consumption, is crucial. Performance analysis serves as a pertinent example of this necessity, as it plays a vital role in pinpointing areas for improvement and optimizing system operations. The research conducted by Bouaouda et al. [23] delves into the performance of energy consumption within a data center utilizing the CloudSim simulator. This study focuses on the strategic placement of containers on hosts and conducts a comparative analysis of two algorithms aimed at reducing energy consumption in cloud systems.

Elasticity is identified as a metric that dynamically adjusts resource consumption in response to varying workload demands, thereby enhancing energy efficiency. The studies by Fontana Nardin et al. [12] and Ferreira et al. [35] investigate strategies for optimizing energy usage by periodically adjusting resource allocation to suit application demands, focusing on models that efficiently manage resources to reduce energy consumption. Valera et al. [42] introduce a simulator designed to deploy and manage diverse networks and heterogeneous devices for running distributed applications based on services or microservices.

Reliability emerges as a crucial consideration, where a malfunction in a single microservice or container could potentially compromise the entire system's availability and reliability. The complexity introduced by microservices and containers can pose challenges in maintaining reliable energy consumption. Lyu et al. [19] propose a k-fault tolerant model aimed at enhancing the reliability of power management systems.

Sustainability in energy usage is of paramount importance in systems that use microservices and containers, attributed to the extensive use of virtual machines and containers. Kumar et al. [24] present a strategy for managing renewable energy-conscious, multiindexed jobs through container services, focusing on maximizing energy efficiency and sustainability within data centers. This approach seeks to reduce energy consumption in data centers, which are significant electricity consumers worldwide, by optimizing the usage of energy resources and reducing the environmental impact, thus underscoring the critical role of sustainability in the long-term efficacy and viability of microservices and container-based systems.

Availability is considered essential for the performance and reliability of systems built on microservices and containers, which may be influenced by energy consumption, especially in high-processing demand scenarios. Jawar et al. [20] advocate for the use of data processing microservices modeling to enhance data availability and introduce services capable of managing energy efficiency.

1) PERFORMANCE

The deployment of microservices within data centers facilitates the efficient utilization of resources, essential for managing fluctuating workloads while maintaining desired throughput and response time. The literature review reveals a focus on such methodologies to guarantee optimal performance. Ranjan et al. [43] discuss strategies for minimizing power consumption in large-scale cloud data centers through container-based virtualization for optimized workflow scheduling, with performance metrics including power usage, workflow execution time, and resource utilization.

2) ELASTICITY

Elasticity in a microservices architecture is crucial, allowing the system to automatically scale its processing, storage, and network capacities in response to workload variations. This capability is particularly vital in distributed environments to ensure system availability and performance during peak

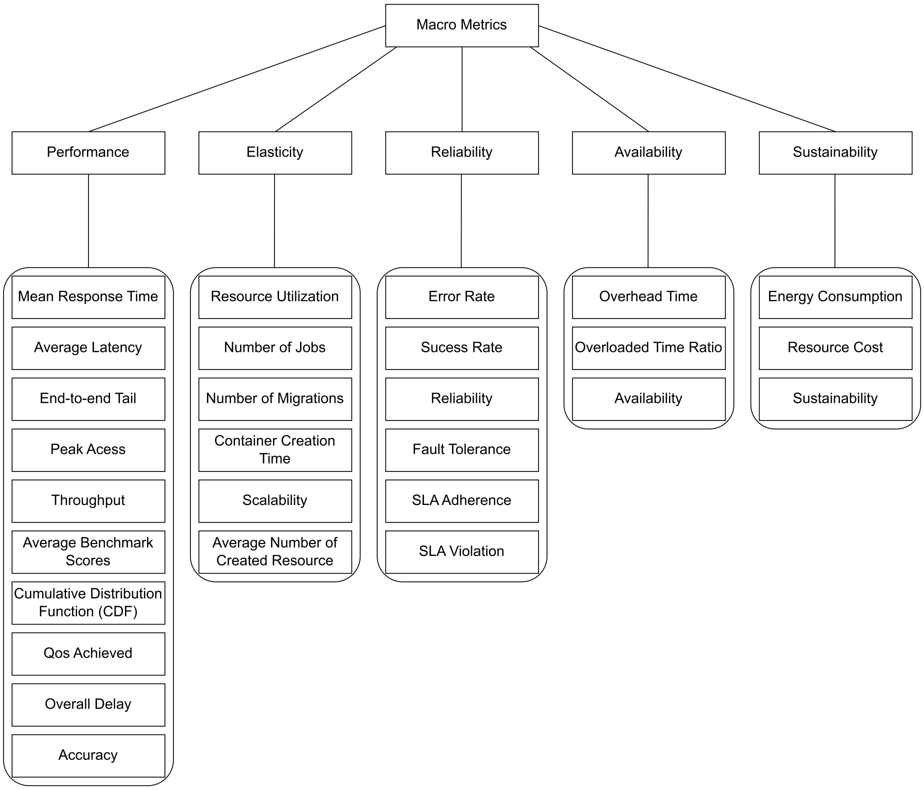


FIGURE 6. Taxonomy of metrics.

demands. Canosa-Reyes et al. [44] present online task allocation strategies to improve service quality, incorporating container selection and capacity distribution among their scheduling policies.

3) RELIABILITY

As technological complexity escalates, the reliability of systems becomes increasingly critical. Li et al. [18] apply discrete Markov theory to adjust redundancy in real time, with the aim of achieving the highest reliability within resource constraints. Lyu et al. [19] introduce a microservices-based architecture for energy management systems to increase the reliability of the system by allowing independent module operation, thus simplifying system complexity.

4) SUSTAINABILITY

The Sustainability metric is a more specific analysis of energy expenditure, addressing the environmental impact of equipment in terms of exergy consumption (available energy) throughout life. Microservice architectures are instrumental in promoting data center sustainability through the strategic use of containers for application and resource grouping. This methodology not only enhances flexibility and scalability, but also facilitates more efficient cloud infrastructure utilization. Kumar et al. [24] propose a renewable energy-based strategy for managing multi-indexed jobs in cloud data centers, aiming to utilize renewable energy resources for greater sustainability and reduced carbon footprint. It is worth mentioning that although only one study explored sustainability as the main metric, all articles addressed energy consumption in their work. Microservices architecture supports sustainability by enabling efficient resource management and adaptability to changing environments. By breaking applications into smaller, independently scalable services, this architecture allows for precise scaling and resource optimization, reducing waste and enhancing cost-efficiency. For instance, stateless components can use cost-effective computing options, minimizing operational costs and environmental impact. Additionally, research explores using microservices for adaptive systems in green buildings, employing machine learning to dynamically adjust systems based on real-time environmental data, thus improving energy efficiency and sustainability [45], [46], [47].

5) AVAILABILITY

Microservices architecture plays a pivotal role in ensuring data availability within the IoT, crucial for autonomous service operation and decision-making. Jawar et al. [20] suggest a microservices model to enhance data availability for energy efficiency management in buildings, thereby enabling more informed energy performance improvements.

C. EVALUATION METHODS

The selection of evaluation methods is crucial to substantiating the validity of research proposals and facilitating their acceptance. These methods not only qualify contributions to the field but also help identify potential research gaps. Figure 7 categorizes the evaluation methods used in the reviewed studies into three main groups: simulation, measurement, and modeling, providing a comprehensive overview of the methodologies applied to assess research results.

Simulation modeling serves as a method to evaluate theoretical propositions and system performance in various scenarios without incurring the costs or risks associated with real-world experiments. This technique offers the advantage of controlled data collection and the ability to investigate complex systems prior to implementation. Nevertheless, inaccuracies in modeling may yield unrealistic or unsatisfactory outcomes, thereby affecting simulation reliability. It is imperative to exercise diligence in the modeling phase to guarantee the precision and dependability of the results. Among the studies reviewed, 56. 8% used simulation modeling.

Measurement offers a more precise and realistic evaluation, reflecting actual system behavior. It enables monitoring of system performance over time under different conditions, helping to identify discrepancies and trends. However, measurement requires access to the system or prototype in question and may require specialized equipment and extensive data collection, making it a more expensive and time-consuming method compared to simulation. Measurement was the method of choice in 37 8% of the articles reviewed.

A minority of performance evaluations employed modeling as their evaluation technique, allowing the simplification and abstraction of complex systems for easier analysis. Only 5.4% of the papers reviewed resorted to modeling.

1) SIMULATION

The categorization of papers under simulation revealed that 10% did not specify the types of simulators used, possibly due to the authors deeming such details as either non-critical or irrelevant to their study's focus.

CloudSim, an open-source cloud simulation framework, was employed in 55% of the papers that disclosed their simulator choice, facilitating the modeling and simulation of cloud computing environments. Khan et al. [15] utilized CloudSim to assess the impact of different allocation policies and migration strategies on data center energy efficiency and performance using Google workload data.

Singh et al. [48] and Al-Moalmi et al. [49] also applied CloudSim to evaluate their proposed methods for load balancing and container placement optimization, respectively, in cloud and edge computing environments.

DRACeo, chosen by 10% of the papers specifying their simulator, is designed to simulate and manage the deployment and scaling of microservice-based applications across various network architectures.

Linpack, used by 5% of the papers, benchmarks the performance of distributed computing systems, focusing on linear algebra computations.

MATLAB 2015b, another tool used in 5% of the simulation-classified papers, supports data analysis, signal processing, and system modeling, as demonstrated by Lyu et al. [19] in their development of an energy management system.

2) MEASUREMENT

Among the standardized measurement articles, 60% did not disclose the tools used, suggesting a potential oversight of the relevance of the tool specification. The Raspberry Pi, used in 20% of these papers, serves as a low-cost, versatile device for data collection and measurement, exemplified by Lennick et al. [41] in their analysis of IoT Linux Docker distributions.

The Grid'5000 testbed, employed in 20% of the studies, provides a scalable and distributed testing environment for evaluating distributed computing system performance, as applied by Xu et al. [50].

3) MODELING

Mathematical models offer a cost-effective and accurate means of evaluating the efficiency and performance of technological systems. These models are crucial in assessing reliability and fault tolerance and ensuring system functionality. Markov Chain Model and K-Fault are two models highlighted in the studies, with applications ranging from optimizing system reliability to improving fault tolerance in service management systems.

D. TYPES OF ARCHITECTURE

The concept of cloud computing emerged in the 1960s, influenced by J.C.R. Licklider who imagined computing in the form of a global network. This concept has gained more and more strength over the years, especially since 2006 with the idea of exchanging hardware for services. Cloud computing is currently one of the most important technologies in the world, expanding into various types of architecture [51]. The distribution of architectures among the reviewed papers is illustrated in Figure 8, with cloud computing dominating at 81. 1%, followed by edge computing at 13. 5% and hybrid computing at 5. 4%. This distribution underscores Cloud Computing's prevalence over Edge and Hybrid Computing in the analyzed literature.

Cloud architectures effectively tackle the challenges associated with large-scale data processing [52]. These architectures encompass a suite of practices and design patterns for developing and deploying applications and services within cloud environments. Cloud computing, along with Fog Computing and Edge Computing, represents distinct models

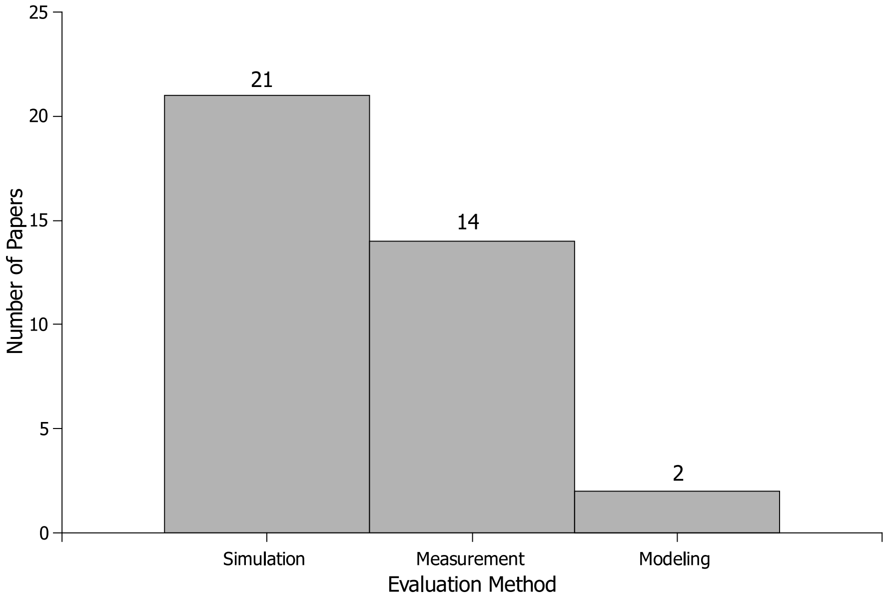


FIGURE 7. General classification of evaluation methods.

of distributed computing architectures that offer users flexible access to computing resources.

Cloud Computing is a service model that allows on-demand access to network storage and computing resources, with users paying only for the resources they consume [53]. Offers benefits such as scalability, flexibility, and cost efficiency, allowing users to easily adjust computing capabilities without the need for direct hardware maintenance and upgrades. This model facilitates only paying for utilized resources, eliminating the need for substantial upfront investments in hardware and infrastructure.

Lennick et al. [41] employ Cloud Computing to evaluate the performance and energy consumption of Linux distributions on IoT devices serving as Docker hosts. Rajan et al. [43] leverage Cloud Computing to support the execution of a genetic algorithm for container consolidation, aimed at enhancing resource utilization and energy efficiency. Gholipour et al. [14] introduce a novel approach to resource management in Cloud Computing, integrating the migration of virtual machines and containers to significantly reduce energy consumption, SLA violations, and migration frequency. Kaur et al. [54] explore the impact of resource consolidation and load balancing within a cloud-based Data Center, focusing on minimizing energy consumption.

Edge computing aims to minimize latency and bandwidth usage by processing data as close to its source as possible [55]. This architecture improves cloud computing by situating computational resources near endpoint devices, such as sensors and IoT devices, making it ideal for real-time data processing applications. Kaur et al. [38] and Asif et al. [56] illustrate the application of Edge Computing in facilitating data and service processing at the network's edge, with containerization playing a pivotal role in the remote programming of IoT devices in the energy sector.

Fog Computing, a decentralized infrastructure, positions computing resources between the data source and the cloud or other data centers, allowing local data processing by edge devices such as IoT sensors [57]. This model enhances latency and energy efficiency, complementing rather than substituting Cloud Computing.

Hybrid Computing merges Cloud and Fog Computing architectures, proving beneficial for deploying containerized services on resource-constrained IoT devices. Raza et al. [36] investigate key factors in deploying containerized services on IoT devices with limited computing and energy resources, aiming to assess the performance and energy consumption of containerized solutions in IoT gateways. The popularization of cloud computing brings light to energy efficiency issues. Data center energy consumption increased from 200 TWh in 2016 to 2,967 TWh in 2030. Data centers need a lot of energy to provide services, which also increases CO_2 emissions [58]. Reducing energy consumption is still an important field of study, as the number of papers covered in this work highlights.

The increasing importance of cloud computing in the context of energy consumption is emphasized by projections indicating significant rises in the sector's energy demands. Current estimates suggest that about 1% of the world's electricity is utilized by cloud computing, a figure expected to surge to as much as 8% by the end of this decade. This increase is driven by the adoption of data-intensive technologies such as augmented reality and smart digital currencies. As the energy demands of cloud computing grow, there is a corresponding surge in academic and industry focus on developing sustainable and energy-efficient

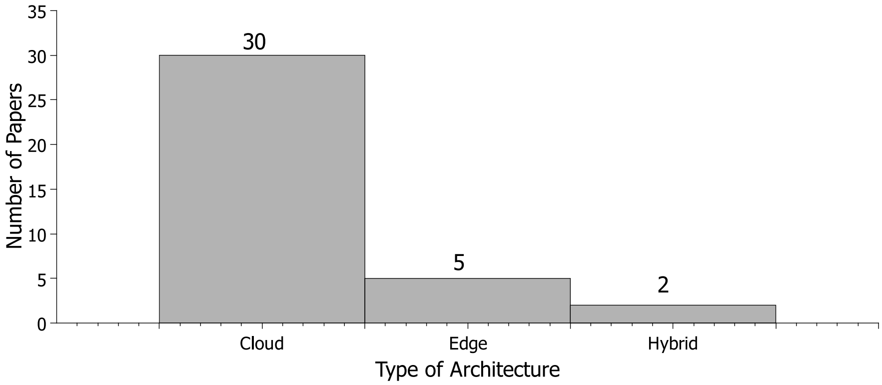


FIGURE 8. Classification of types of architecture.

solutions within cloud infrastructures. The literature over the past decade reflects a pivotal shift towards enhancing the energy efficiency of cloud technologies, highlighting efforts to reduce both the economic costs and environmental impacts associated with their expanding use [59], [60], [61].

E. ORCHESTRATOR

The distribution of orchestrators within the reviewed articles is shown in Figure 9. The analysis indicates that 43% of the studies did not specify the orchestrator type used. Among the papers that provided this detail, Kubernetes and Docker Swarm emerged as the primary container orchestrators, illustrating the prevalent use of these tools in managing containerized services and applications.

Container orchestrators are instrumental tools for managing the execution of containers across systems. These tools facilitate the creation, deployment, and management of containerized applications with increased efficiency. Moreover, container orchestrators monitor resource consumption and can dynamically scale applications in response to fluctuating demands.

The Kubernetes orchestrator, an open source platform, is designed to automate the deployment, scaling, and operations of containerized applications across clusters. Saengkaenpetch et al. [16] employ Kubernetes to orchestrate the necessary container workloads. Docker Swarm, another container management system, enables the creation and management of container clusters, streamlining application scaling, and enhancing service availability. Lennick et al. [41] utilize Docker to scrutinize the performance and energy usage of Linux distributions in conjunction with IoT devices.

1) KUBERNETES

Within the studies classified as orchestrators, 17.9% chose Kubernetes. This platform is recognized for its ability to automate the scaling, deployment, and management of containerized applications. Lyu et al. [19] apply Kubernetes to oversee container health within a power management framework, utilizing liveness and readiness probes to assess container health and readiness to handle requests, respectively, aiming to bolster system reliability.

2) DOCKER SWARM

Analysis reveals that 35.9% of the orchestrator-related studies opted for Docker Swarm, indicating its preferred status for managing expansive distributed systems. Zhang et al. [62] concentrate on evaluating container energy consumption performance, employing Docker Swarm for experimental analyses to compare their proposed strategy against two Docker-native strategies for optimizing energy usage.

F. DEPLOYMENT MODEL

Figure 10 outlines the classification results for deployment models, showing that 83.8% of the reviewed papers did not detail the deployment model type, suggesting a general lack of emphasis on this information among the studies' authors. The deployment model concept pertains to the cloud configuration utilized within the investigated works, categorizing cloud types into public, private, and hybrid models. These models may be employed alone or in combination, as illustrated in the chart, reflecting the diversity in cloud deployment strategies adopted across the studies.

1) PUBLIC

Luo et al. [63] illustrate that while traditional fog computing architectures, centered around a data center and multiple nodes, face challenges in adapting to the evolution of private cloud development, deploying a multi-cloud architecture utilizing containers within the public cloud offers a solution that enhances operational flexibility, agility, and scalability for power balancing. The integration of microservices and containers in the public cloud emerges as an efficient approach for organizations aiming for increased adaptability and efficacy in the development and operation of applications.

2) PRIVATE

The private cloud emerges as a strategic option for enterprises prioritizing data privacy and security, providing access

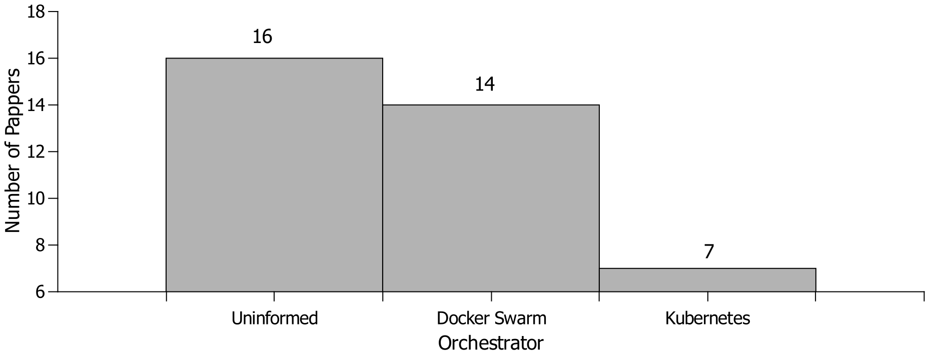


FIGURE 9. Classification of orchestrator types.

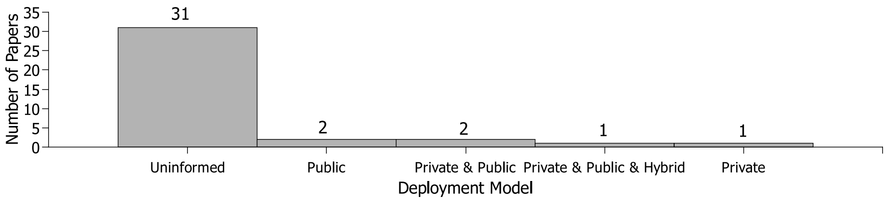


FIGURE 10. Classification of deployment model.

to technologies and services comparable to those in the public cloud but within a proprietary and organization-controlled environment. Fontana de Nardin et al. [12] leverage the private cloud to develop and assess a prototype for a microservices-based application orchestrated using Docker Swarm. Piraghaj et al. [64] concentrate on server energy efficiency enhancement through a framework that consolidates containers within virtual machines, employing two cloud deployment models. The private cloud model focuses on reducing energy consumption through specific algorithms, while the public cloud model aims to decrease the total virtual machine leasing time.

3) HYBRID

Hybrid cloud deployment merges the strengths of both public and private clouds, offering businesses the public cloud's agility and flexibility alongside the private cloud's security and control. Asif et al. [56] analyze the container implementation architecture for energy sector applications, particularly in state-of-the-art substation automation systems. The benefits of container technology include its applicability across various cloud models, including hybrid, private, and public clouds.

V. CROSS-CORRELATION CLASSIFICATION RESULTS

This section elucidates the frequency of studies based on the interrelationship between two distinct criteria, displaying the number of studies at the intersections of the x and y axes within graphs. This multidimensional analysis, considering various criteria simultaneously, aids in identifying prevalent research domains and potential gaps for future exploration.

A. ARCHITECTURE X DEPLOYMENT MODEL

Figure 11 delineates the interplay between Architectural Types and Deployment Model classifications, highlighting that cloud computing is predominantly applied across different deployment models, including private, public, and hybrid. The y-axis categorizes the types of architectures, such as Cloud Computing, Edge Computing, and Hybrid Computing, while the x-axis details the Deployment Model, encompassing Private, Public, Hybrid, and unspecified models.

Edge computing's application is notably less frequent compared to cloud computing, with only five studies mentioning it. Edge computing is associated with private, public, and hybrid deployment models, whereas other uses remain undetailed. For fog and cloud computing, public deployments account for 50% of the studies, with the remainder unspecified. The observed lack of detailed deployment model descriptions suggests that the studies prioritize more pertinent criteria, thus diverting attention to other research areas. For instance, Bouaouda et al. [23] propose a method to forecast energy consumption in cloud data centers.

B. ARCHITECTURE X ORCHESTRATOR

Figure 12 delineates the intersection of classifications concerning Architectural Types and Orchestrators. On this graph, the x-axis categorizes the orchestrators in operation: Docker Swarm, Kubernetes; both Docker Swarm and Kubernetes; and those unspecified. The y-axis outlines the Architectural Types utilized, including Cloud, Edge, and Hybrid Computing. Notably, the Cloud Computing architecture emerged as the predominant choice among the studies, featuring in 29 selected works and across all orchestrator options.

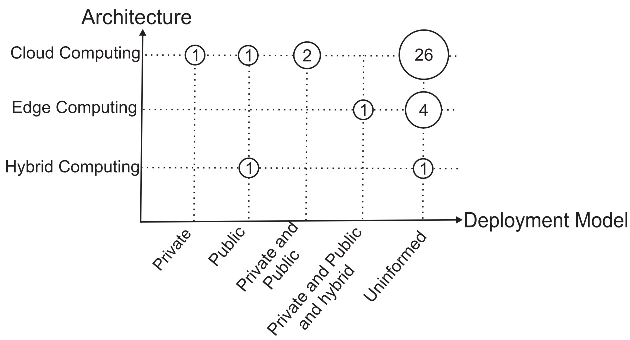


FIGURE 11. Architecture x Deployment Model.

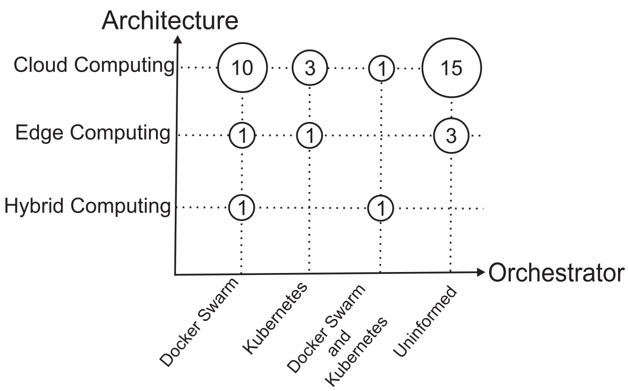


FIGURE 12. Architecture x Orchestrator.

Edge computing's deployment with Docker Swarm and Kubernetes is observed to be comparable, with a noted absence of details regarding the orchestrator use in other studies. The adoption of computing in fog and cloud environments is minimal compared to other architectures. Edge, fog, and cloud computing are less frequently explored in the literature when juxtaposed with the extensively accepted Cloud Computing architecture. The study by Khan et al. [15] exemplifies a preference for Cloud Computing, given its relevance to the context in which the proposed resource consolidation technique is implemented.

C. EVALUATION METHOD X ARCHITECTURE

Figure 13 elucidates the relationship between the Evaluation Methods and Architectural Types classifications. On this graph, the x-axis categorizes the Architectural Types, including Hybrid Computing, Edge Computing, and Cloud Computing. Concurrently, the y-axis highlights the Evaluation Methods utilized: Simulation, Measurement, and Modeling. It is evident that simulation emerged as the predominant evaluation method, closely followed by measurement, with both methods spanning across all presented architectural types. Conversely, modeling was scarcely represented in the collected works, finding application solely within cloud computing contexts.

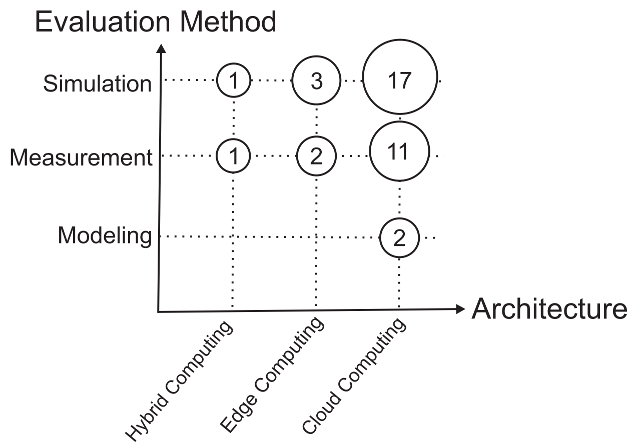


FIGURE 13. Evaluation Method x Architecture.

The prevalence of the simulation evaluation method is particularly notable within cloud computing architectures, underscoring a significant discrepancy in its application across edge computing and fog and cloud computing architectures when compared to cloud computing. The research conducted by Canosa-Reyes et al. [44] exemplifies the utilization of simulation to facilitate the assessment of proposed techniques within a controlled setting, enabling the exploration of various scenarios and the systematic collection of performance data. Similarly, when measurement serves as the evaluation method, cloud computing attracts a substantial concentration of studies, while other architectural types experience a marked decline in attention.

D. EVALUATION METHOD X ORCHESTRATOR

Figure 14 shows the relationship between Evaluation Methods and Orchestrators. In this chart, the x-axis categorizes orchestrators: Docker Swarm, Kubernetes; both Docker Swarm and Kubernetes; and those unspecified. The y-axis highlights the Assessment Methods used: Simulation, Measurement, and Modeling. Simulation stood out as the most used evaluation method, followed by measurement. Modeling was poorly represented with only two works using the Kubernets orchestrator. The vast majority of works exploring simulation and measurement did not specify which orchestrator they used. The Docker Swarm orchestrator was very prominent in simulation and measurement work. The Kubernetes orchestrator was used for modeling and some measurement work, but was not used in isolation for simulation work. Only the work by Raza et al. [36] presented a mixed orchestrator approach, using Docker Swarm in conjunction with Kubernetes. Raza et al. sought to establish guidelines and identify critical factors for the deployment of containerized services on resource-constrained IoT devices. In the work Raza et al. the Docker Swarm and Kubernetes orchestrators were tested and compared on an IoT gateway testbed through simulation.

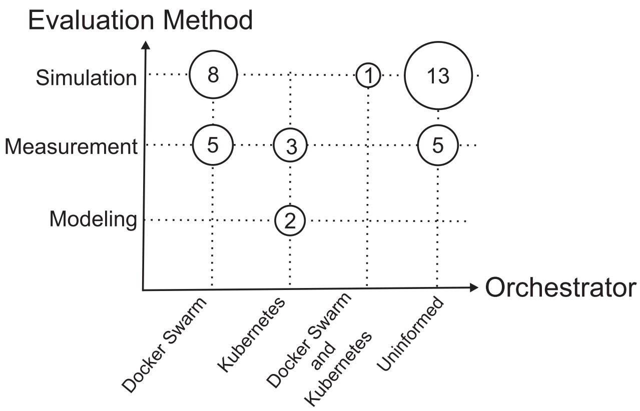


FIGURE 14. Evaluation Method x Orchestrator.

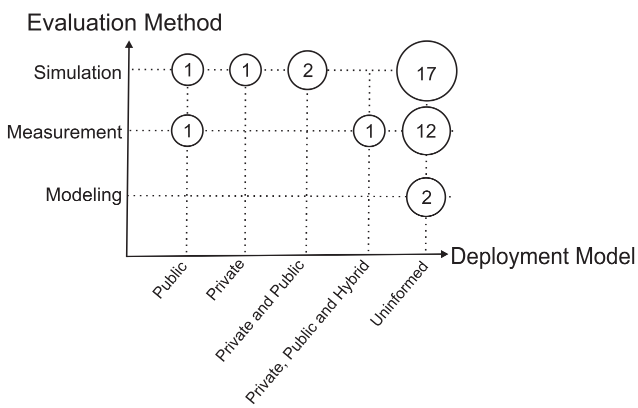


FIGURE 15. Evaluation Method x Deployment Model.

E. EVALUATION METHOD X DEPLOYMENT MODEL

Figure 15 delineates the intersection between Evaluation Methods and Deployment Model classifications. The Deployment Model is categorized along the x-axis, encompassing Private, Public, Private and Public, Private and Public and Hybrid, and Unspecified models. Concurrently, the y-axis showcases the Evaluation Method employed, which includes Simulation, Measurement, and Modeling. From the graphical representation, it is clear that simulation emerges as the most prevalent method within the field, surpassing the other methods in terms of discussion and application. Measurement also enjoys considerable application across various studies, whereas modeling receives less emphasis in comparison.

Simulation, as an evaluation technique, is extensively utilized across the studies, though specific details regarding its deployment in various models are often not explicitly stated. The limited number of studies that do specify its application reveal its usage across diverse deployment settings, including private, public, and combinations of private and public models. Measurement's application appears to be less prevalent in private deployments and mixed private and public contexts, while modeling is not distinctly attributed to any specific Deployment Model in the analyzed works.

Most of the simulation works used the CloudSim simulator (eleven works). Some other simulators such as BookSim (one job), Accel-Sim v1 (one job) and DRACeo (two jobs) were used on a smaller scale. In measurement and modeling, most studies did not report which tools they used. Only three measurement works reported the use of Raspberry Pi for their tests. Therefore, only three measurement studies validated their proposals in a testbed environment. Most studies did not report which tools they used. However, the CloudSim simulator was the most popular among the works that informed the simulation tools.

F. MACRO METRICS X ARCHITECTURE

Figure 16 elucidates the relationship between Macro Metrics and Architectural Types. This graph allows for an examination of how certain Macro Metrics are prioritized across various architectural types, including Hybrid Computing, Edge Computing, and Cloud Computing, as denoted on the x-axis. The y-axis specifies the metrics under consideration, namely Performance, Elasticity, Scalability, Reliability, Sustainability, and Availability. A notable emphasis on Performance and Elasticity metrics is observed, alongside a predominant adoption of the Cloud Computing architecture across the studies.

Performance and Elasticity metrics emerged as focal points among the studies, underscoring a significant interest in assessing these aspects within the context of Energy Consumption in Microservices. Xu et al. [50] exemplify this focus through their work on performance evaluation, aiming to develop an integrated strategy that addresses both energy conservation and performance optimization. Although other metrics such as Scalability, Reliability, Sustainability, and Availability were explored, they were less prevalent compared to Performance and Elasticity.

The analysis further reveals a pronounced preference for the Cloud Computing architecture across all metrics, attributed to its cost-effectiveness and ease of use. A total of 30 studies demonstrated the utilization of Cloud Computing architecture. Conversely, studies incorporating Edge Computing and Hybrid Computing architectures were primarily associated with Performance, Elasticity, and Availability metrics, indicating a more limited application of these architectural types within the scope of the analyzed works.

G. MACRO METRICS X EVALUATION METHOD

Figure 17 illustrates the intersection between Macro Metrics and Evaluation Methods, providing insight into how different metrics are applied across various assessment techniques. The x-axis categorizes the Evaluation Methods into Modeling, Measurement, and Simulation, while the y-axis lists the Macro Metrics, including Performance, Elasticity, Scalability, Reliability, Sustainability, and Availability. The distribution indicates a notable prevalence of Performance and Elasticity metrics compared to others, highlighting their significant application across the studies.

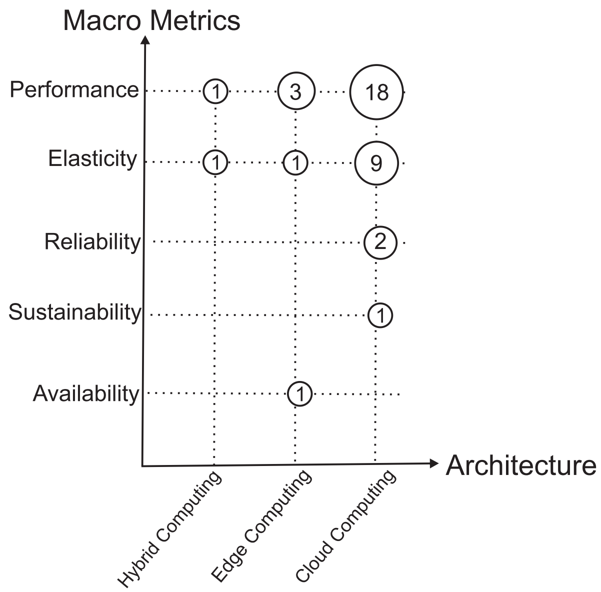


FIGURE 16. Macro Metrics x Architecture.

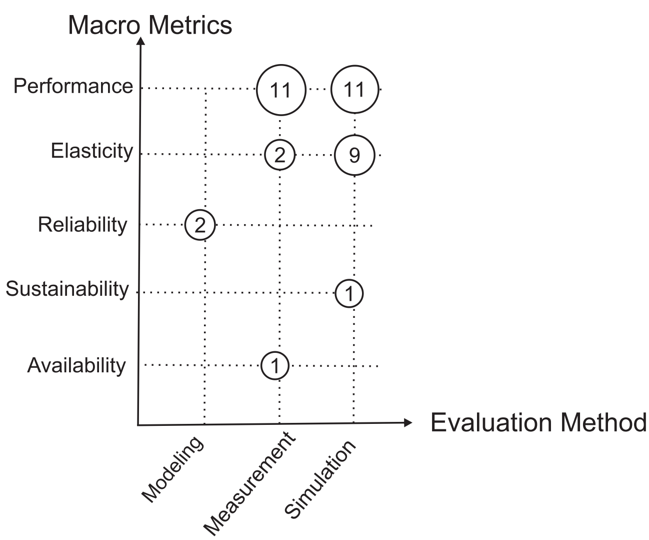


FIGURE 17. Macro Metrics x Evaluation Method.

A particular focus on Performance metrics is evident, with Zhang et al. [62] leveraging Performance and Measurement as central elements in their strategy for optimal container placement. Performance metrics emerge as the most frequently utilized, closely associated with Measurement and Simulation as preferred methods of evaluation. This trend underscores the studies' inclination towards quantifying and simulating performance within microservices environments, especially concerning energy consumption.

Elasticity metrics also garnered substantial attention, predominantly employing the Simulation method for evaluation. This indicates a keen interest in exploring the adaptability of microservices architectures to dynamic environmental conditions and their impact on energy efficiency.

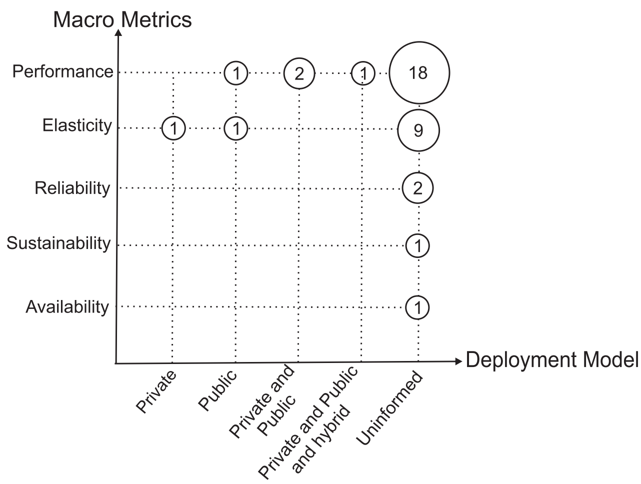


FIGURE 18. Macro Metrics x Deployment Model.

H. MACRO METRICS X DEPLOYMENT MODEL

Figure 18 elucidates the interrelation between Macro Metrics and Deployment Model classifications, showcasing the deployment models alongside the various metrics assessed within the studies. The x-axis details the Deployment Model types, including Private, Public, combinations of Private and Public, Private and Public and Hybrid, as well as those unspecified. Conversely, the y-axis outlines the metrics evaluated: Performance, Elasticity, Scalability, Reliability, Sustainability, and Availability. Upon analysis, it is apparent that a significant portion of the studies did not disclose their specific Deployment Model, with Performance and Elasticity metrics emerging as the most frequently employed across the works.

In the realm of Performance metrics, there is a notable trend towards their application across different Deployment Models, although many studies refrain from explicitly stating their chosen model. This pattern similarly extends to studies utilizing the Elasticity metric, where explicit mention of Deployment Models is often omitted. Works employing other metrics largely bypassed specifying their Deployment Model, thus being categorized as unspecified.

Notably, some studies focusing on Performance metrics did delineate their Deployment Models, including Public, combinations of Private and Public, as well as Private and Public and Hybrid models. Conversely, studies emphasizing Elasticity metrics predominantly featured deployments blending Private and Public models. In contrast, studies incorporating metrics beyond Performance and Elasticity largely neglected to specify their Deployment Model, resulting in their classification as unspecified.

I. MACRO METRICS X ORCHESTRATOR

Figure 19 delineates the correlation between Macro Metrics and Orchestrator classifications, providing insights into the utilization of orchestrators across various metrics. The x-axis categorizes the orchestrators, including Kubernetes, Docker

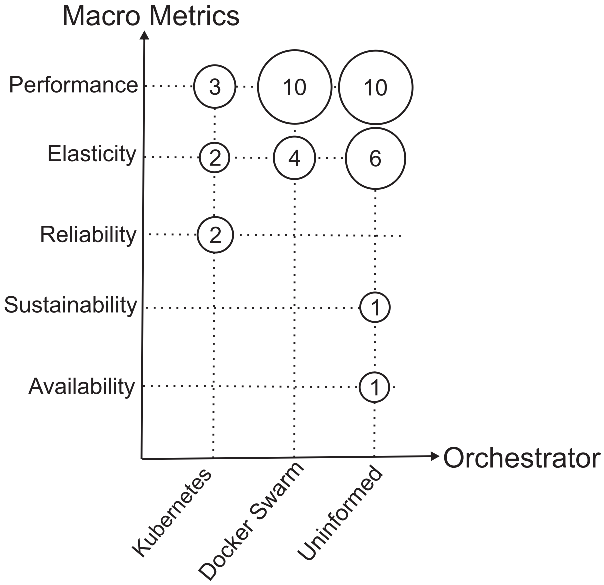


FIGURE 19. **Subcontext x Orchestrator.**

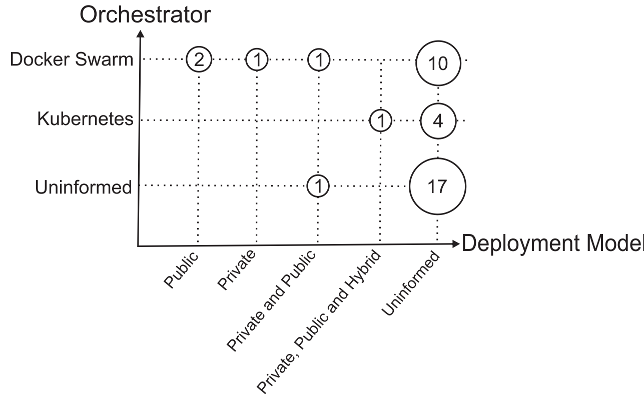


FIGURE 20. **Orchestrator x Deployment Model.**

Swarm, and those unspecified, while the y-axis lists the metrics employed across the studies: Performance, Elasticity, Scalability, Reliability, Sustainability, and Availability. The analysis reveals that studies focusing on Sustainability and Availability metrics either did not associate with a specific orchestrator or did not disclose this detail. Conversely, Performance and Elasticity metrics were more frequently explored within the works.

Studies emphasizing Performance metrics predominantly utilized the Docker Swarm orchestrator, with a significant number also employing Kubernetes, and an equal proportion of studies refraining from specifying their orchestrator choice. Regarding the Elasticity and Scalability metrics, a subset of the studies engaged an orchestrator for their evaluation. In contrast, all studies assessing Reliability metrics exclusively opted for Kubernetes as their orchestrator. Meanwhile, studies investigating Sustainability and Availability metrics did not indicate the orchestrator employed, underscoring a gap in the disclosure or application of orchestrators within these particular areas of research.

J. ORCHESTRATOR X DEPLOYMENT MODEL

Figure 20 elucidates the correlation between Orchestrator and Deployment Model classifications, illustrating the deployment models and orchestrators as identified in the studies. The x-axis of the graph categorizes the Deployment Model Types as Private, Public, combinations of Private and Public, Private and Public and Hybrid, along with those unspecified. Meanwhile, the y-axis lists the Orchestrators employed: Kubernetes, Docker Swarm, and those studies that did not specify an orchestrator. Analysis of the chart reveals a notable preference for the Docker Swarm orchestrator and a significant number of works that omitted the specification of both Orchestrator and Deployment Model Type.

The Docker Swarm orchestrator was associated with a range of Deployment Models, including Public, Private, and combinations of Private and Public. However, a substantial portion of the studies utilizing Docker Swarm did not specify their Deployment Model. In contrast, the Kubernetes orchestrator was predominantly connected to Deployment Models that blend Private and Public and Hybrid approaches, with several other studies not disclosing their Deployment Model.

A majority of the studies did not explicitly mention their choice of Orchestrator or Deployment Model Type. There were instances where the studies failed to indicate the orchestrator utilized but provided details on their Deployment Model, typically a blend of Private and Public models. This pattern suggests a discernible gap in the reporting or application of specific orchestrators and deployment models within the microservices architecture domain.

VI. RESEARCH CHALLENGES AND DIRECTIONS

This section outlines the challenges and prospective research directions derived from an in-depth analysis of 37 papers focusing on microservices and energy consumption. It is organized around key challenges identified in this study, including energy consumption, resource management, container migration, and task scheduling.

A. ENERGY CONSUMPTION

The proliferation of services and the demand for constant availability significantly elevate energy consumption levels. Effective monitoring and optimization of energy use are crucial for enhancing sustainability. The continuous operation of container orchestrators and resource management algorithms, coupled with an increase in server and service counts, escalates energy demands in data centers [20], [22]. Elevated energy consumption further complicates issues such as cooling costs, overloads, and energy expenses. Although techniques like load balancing and efficient task scheduling exist, they may compromise service quality and diminish cloud application performance [22], [44]. Consequently, there is a pressing need for innovative energy reduction strategies tailored for microservice architectures.

B. RESOURCE MANAGEMENT

The exponential growth in data generation exacerbates the volumes of service requests in data centers, leading to processing delays and prolonged response times [22], [48]. Inefficient data management can result in service interruptions, unfulfilled requests, and increased energy consumption. Effective management of data center resources, including CPU, RAM, and storage, is essential to meet service requests without resource wastage. Microservices architectures offer improved resource management through service decomposition, yet adjusting elasticity presents challenges that require careful consideration of various factors [12], [48], [49]. Efficient and consistent resource management is vital for adapting applications to fluctuating load demands.

C. CONTAINER MIGRATION

Container migration serves as a strategy for managing workloads in cloud applications. However, migrations can incur significant energy and performance costs, often overlooked in many models. Resource allocation and consolidation, viewed as multidimensional bin-packing problems, gain complexity with host heterogeneity and varied resource types [14], [15]. Poorly executed migrations can lead to resource contention, decreased performance, and increased energy costs [15], [22].

D. TASK SCHEDULING

Efficient task scheduling in microservice architectures remains a formidable challenge due to the variability of the traffic flow from cloud applications. Scheduling tasks in large, dynamic, and elastic computing environments requires addressing priority levels and computational demands without compromising service performance [43], [44]. Inappropriate scheduling can escalate power consumption and breach Service Level Agreements (SLAs) [65], [66].

Furthermore, while performance metrics have received significant attention, essential metrics such as reliability, availability, and sustainability warrant further exploration as potential research avenues in the realm of microservices energy consumption.

VII. CONCLUSION

This document delineates the results of a systematic review focused on energy consumption within microservice architectures, providing significant information on current research trends. Predominantly, the studies examined have concentrated on performance metrics, a focus that aligns with the critical importance of performance in ensuring the success of microservices applications. Notably, elasticity metrics have emerged as the second most investigated aspect, with nine specific studies dedicating their efforts to this area. The emphasis on elasticity is rational, considering its role in facilitating effective resource management, thereby contributing to the reduction of energy consumption in microservices environments. Despite this focus, it is observed that explorations into reliability, availability, and sustainability metrics are less prevalent, presenting potential avenues for future research.

The evaluation of microservice architectures typically employs one of three primary methods: simulation, measurement, and modeling. Simulation stands out as the most utilized technique, largely supported by the capabilities of the CloudSim simulator, underscoring its utility in cloud architecture analysis. Measurement is also a commonly adopted method, though modeling has been applied less frequently, with only two studies employing this approach. Despite its relatively limited use, modeling is recommended for future research due to its cost-effectiveness and high level of abstraction, despite offering less precision than empirical experiments.

Cloud computing has been the main architectural focus, with edge and hybrid architectures receiving comparatively less attention. Moreover, the discussions predominantly revolve around architectural types, with scant details on Deployment Models. Similarly, the choice of orchestrator is not thoroughly explored, though, when mentioned, Docker Swarm appears to be preferred over Kubernetes. Given the overarching theme of energy consumption in microservices, data centers emerge as the most commonly studied context, with some investigations extending into the Internet of Things (IoT) domain.

This study acknowledges three primary threats to its validity: (i) the potential for research questions to not encompass all pertinent studies on microservices power consumption, (ii) the risk of selection bias possibly omitting relevant research, and (iii) the subjective nature of data extraction. Efforts to mitigate these threats included a comprehensive research strategy and collaborative classification by six researchers.

Future endeavors will aim to dive into the underexplored challenges related to controlling energy consumption in microservices, along with conducting additional systematic reviews within the microservices field.

REFERENCES

[1] I. Fé, R. Matos, J. Dantas, C. Melo, T. A. Nguyen, D. Min, E. Choi, F. A. Silva, and P. R. M. Maciel, ‘‘Performance-cost trade-off in autoscaling mechanisms for cloud computing,’’ *Sensors*, vol. 22, no. 3, p. 1221, Feb. 2022.

[2] L. Feitosa, L. Santos, G. Goncalves, T. A. Nguyen, J.-W. Lee, and F. A. Silva, ‘‘Internet of Robotic Things: A comparison of message routing strategies for cloud-fog computing layers using m/m/c/k queuing networks,’’ in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2021, pp. 2049–2054.

[3] L. Santos, B. Cunha, I. Fé, M. Vieira, and F. A. Silva, ‘‘Data processing on edge and cloud: A performability evaluation and sensitivity analysis,’’ *J. Netw. Syst. Manag.*, vol. 29, no. 3, p. 27, Jul. 2021.

[4] F. A. Silva, I. Fé, and G. Gonçalves, ‘‘Stochastic models for performance and cost analysis of a hybrid cloud and fog architecture,’’ *J. Supercomput.*, vol. 77, no. 2, pp. 1537–1561, Feb. 2021.

[5] F. A. Gomes, P. A. Rego, F. A. M. Trinta, W. Viana, F. A. Silva, J. A. de Macêdo, and J. N. de Souza, ‘‘A study about the impact of encryption support on a mobile cloud computing framework,’’ in *Proc. CLOSER*, 2020, pp. 400–407.

[6] P. T. Endo, T. Lynn, F. A. Silva, D. Sadok, J. Kelner, G. L. Santos, and D. Gomes, "The Internet of Things for healthcare: Optimising e-health system availability in the fog and cloud," *Int. J. Comput. Sci. Eng.*, vol. 21, no. 4, p. 615, 2020.

[7] T. Pinheiro, F. A. Silva, I. Fé, D. Oliveira, and P. Maciel, "Performance and resource consumption analysis of elastic systems on public clouds," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 2115–2120.

[8] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, Jun. 2009.

[9] Gartner. (Oct. 2022). *Gartner Forecasts Worldwide Public Cloud End-User Spending to Reach Nearly $600 Billion in 2023*. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2022-10-31-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-nearly-600-billion-in-2023#:~:text=Worldwide%20end%2Duser%20spending%20on,latest%20forecast%20from%20Gartner%2C%20Inc

[10] N. Dragoni, S. Giallorenzo, A. L. Lafuente, M. Mazzara, F. Montesi, R. Mustafin, and L. Safina, "Microservices: Yesterday, today, and tomorrow," in *Microservices: Yesterday, Today, and Tomorrow*, 2017, pp. 195–216.

[11] X. Larrucea, I. Santamaria, R. Colomo-Palacios, and C. Ebert, "Microservices," *IEEE Softw.*, vol. 35, no. 3, pp. 96–100, May 2018.

[12] I. F. D. Nardin, R. D. R. Righi, T. R. L. Lopes, C. A. da Costa, H. Y. Yeom, and H. Köstler, "On revisiting energy and performance in microservices applications: A cloud elasticity-driven approach," *Parallel Comput.*, vol. 108, Dec. 2021, Art. no. 102858, doi: 10.1016/j.parco.2021.102858.

[13] M. Liu, S. Peter, A. Krishnamurthy, and P. M. Phothilimthana, "E3: Energy-efficient microservices on SmartNIC-accelerated servers," in *Proc. USENIX Annu. Tech. Conf. (USENIX ATC)*, 2019, pp. 363–378.

[14] N. Gholipour, E. Arianyan, and R. Buyya, "A novel energy-aware resource management technique using joint VM and container consolidation approach for green computing in cloud data centers," *Simul. Model. Pract. Theory*, vol. 104, Nov. 2020, Art. no. 102127.

[15] A. A. Khan, M. Zakarya, R. Buyya, R. Khan, M. Khan, and O. Rana, "An energy and performance aware consolidation technique for containerized datacenters," *IEEE Trans. Cloud Comput.*, vol. 9, no. 4, pp. 1305–1322, Oct. 2021, doi: 10.1109/TCC.2019.2920914.

[16] K. Saengkaenpetch and C. Aswakul, "Cloud-based smart energy framework for accelerated data analytics with parallel computing of orchestrated containers: Study case of CU-BEMS," in *Proc. 3rd Int. Conf. Adv. Inf. Sci. Syst.* New York, NY, USA: Association for Computing Machinery, Nov. 2021, pp. 1–6, doi: 10.1145/3503047.3503088.

[17] C.-C. Lin, J.-J. Chen, P. Liu, and J.-J. Wu, "Energy-efficient core allocation and deployment for container-based virtualization," in *Proc. IEEE 24th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2018, pp. 93–101, doi: 10.1109/PADSW.2018.8644537.

[18] Z. Li, H. Wei, Z. Lyu, and C. Lian, "Kubernetes-container-cluster-based architecture for an energy management system," *IEEE Access*, vol. 9, pp. 84596–84604, 2021.

[19] Z. Lyu, H. Wei, X. Bai, and C. Lian, "Microservice-based architecture for an energy management system," *IEEE Syst. J.*, vol. 14, no. 4, pp. 5061–5072, Dec. 2020, doi: 10.1109/JSYST.2020.2981095.

[20] M. A. Jarwar, S. Ali, and I. Chong, "A microservices model to enhance the availability of data for building energy efficiency management services," *Energies*, vol. 12, no. 3, p. 360, Jan. 2019.

[21] M. Khairy, A. Alawneh, A. Barnes, and T. G. Rogers, "SIMR: Single instruction multiple request processing for energy-efficient data center microservices," in *Proc. 55th IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Oct. 2022, pp. 441–463.

[22] H. H. A. Valera, M. Dalmau, P. Roose, J. Larracoechea, and C. Herzog, "An energy saving approach: Understanding microservices as multidimensional entities in P2P networks," in *Proc. 36th Annu. ACM Symp. Appl. Comput.*, 2021, pp. 69–78.

[23] A. Bouaouda, K. Afdel, and R. Abounacer, "Forecasting the energy consumption of cloud data centers based on container placement with ant colony optimization and bin packing," in *Proc. 5th Conf. Cloud Internet Things (CIoT)*, Mar. 2022, pp. 150–157, doi: 10.1109/CIoT53061.2022.9766522.

[24] N. Kumar, G. S. Aujla, S. Garg, K. Kaur, R. Ranjan, and S. K. Garg, "Renewable energy-based multi-indexed job classification and container management scheme for sustainability of cloud data centers," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 2947–2957, May 2019, doi: 10.1109/TII.2018.2800693.

[25] H. Vural, M. Koyuncu, and S. Guney, "A systematic literature review on microservices," in *Computational Science and Its Applications—ICCSA*, Trieste, Italy. Springer, 2017, pp. 203–217.

[26] J. Ghofrani and D. Lübke, "Challenges of microservices architecture: A survey on the state of the practice," in *Proc. ZEUS*, 2018, pp. 1–8.

[27] M. Viggiato, R. Terra, H. Rocha, M. Tulio Valente, and E. Figueiredo, "Microservices in practice: A survey study," 2018, *arXiv:1808.04836*.

[28] J. Soldani, D. A. Tamburri, and W.-J. Van Den Heuvel, "The pains and gains of microservices: A systematic grey literature review," *J. Syst. Softw.*, vol. 146, pp. 215–232, Dec. 2018.

[29] D. Yu, Y. Jin, Y. Zhang, and X. Zheng, "A survey on security issues in services communication of Microservices-enabled fog applications," *Concurrency Comput., Pract. Exper.*, vol. 31, no. 22, p. e4436, Nov. 2019.

[30] P. Di Francesco, P. Lago, and I. Malavolta, "Architecting with microservices: A systematic mapping study," *J. Syst. Softw.*, vol. 150, pp. 77–97, Apr. 2019.

[31] M. Waseem, P. Liang, and M. Shahin, "A systematic mapping study on microservices architecture in DevOps," *J. Syst. Softw.*, vol. 170, Dec. 2020, Art. no. 110798.

[32] A. Hannousse and S. Yahiouche, "Securing microservices and microservice architectures: A systematic mapping study," *Comput. Sci. Rev.*, vol. 41, Aug. 2021, Art. no. 100415.

[33] S. N. A. Jawaddi, M. H. Johari, and A. Ismail, "A review of microservices autoscaling with formal verification perspective," *Softw., Pract. Exper.*, vol. 52, no. 11, pp. 2476–2495, Nov. 2022.

[34] W. Silva-de-Souza, A. Iranfar, A. Bráulio, M. Zapater, S. Xavier-de-Souza, K. Olcoz, and D. Atienza, "Containergy—A container-based energy and performance profiling tool for next generation workloads," *Energies*, vol. 13, no. 9, p. 2162, May 2020.

[35] L. C. B. C. Ferreira, A. D. R. Borchardt, G. D. S. Cardoso, D. A. Mendes Lemes, G. R. D. Sousa, F. B. Neto, E. R. de Lima, G. Fraidenraich, P. Cardieri, and L. G. P. Meloni, "Edge computing and microservices middleware for home energy management systems," *IEEE Access*, vol. 10, pp. 109663–109676, 2022, doi: 10.1109/ACCESS.2022.3214229.

[36] S. M. Raza, J. Jeong, M. Kim, B. Kang, and H. Choo, "Empirical performance and energy consumption evaluation of container solutions on resource constrained IoT gateways," *Sensors*, vol. 21, no. 4, p. 1378, Feb. 2021.

[37] Md. F. Bari, R. Boutaba, R. Esteves, L. Z. Granville, M. Podlesny, M. G. Rabbani, Q. Zhang, and M. F. Zhani, "Data center network virtualization: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 909–928, 2nd Quart., 2013, doi: 10.1109/SURV.2012.090512.00043.

[38] K. Kaur, T. Dhand, N. Kumar, and S. Zeadally, "Container-as-a-Service at the edge: Trade-off between energy efficiency and service availability at fog nano data centers," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 48–56, Jun. 2017.

[39] D. Patel, M. K. Patra, and B. Sahoo, "Energy efficient genetic algorithm for container consolidation in cloud system," in *Proc. 7th Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Feb. 2020, pp. 1066–1071, doi: 10.1109/SPIN48934.2020.9071005.

[40] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, "Internet of Things (IoT) communication protocols: Review," in *Proc. 8th Int. Conf. Inf. Technol. (ICIT)*, May 2017, pp. 685–690, doi: 10.1109/ICITECH.2017.8079928.

[41] D. Lennick, A. Azim, and R. Liscano, "A microservice-based architecture for performance and energy benchmarking of docker-host Linux distributions on Internet-of-Things devices," in *Proc. 22nd IEEE Int. Conf. Ind. Technol. (ICIT)*, vol. 1, Mar. 2021, pp. 705–711.

[42] H. H. A. Valera, M. Dalmau, P. Roose, J. Larracoechea, and C. Herzog, "DRACeo: A smart simulator to deploy energy saving methods in microservices based networks," in *Proc. IEEE 29th Int. Conf. Enabling Technol., Infrastructure Collaborative Enterprises (WETICE)*, Sep. 2020, pp. 94–99, doi: 10.1109/WETICE49692.2020.00026.

[43] R. Ranjan, I. S. Thakur, G. S. Aujla, N. Kumar, and A. Y. Zomaya, "Energy-efficient workflow scheduling using container-based virtualization in software-defined data centers," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7646–7657, Dec. 2020, doi: 10.1109/TII.2020.2985030.

[44] R. M. Canosa-Reyes, A. Tchernykh, J. M. Cortés-Mendoza, B. Pulido-Gaytan, R. Rivera-Rodriguez, J. E. Lozano-Rizk, E. R. Concepción-Morales, H. E. C. Barrera, C. J. Barrios-Hernandez, F. Medrano-Jaimes, A. Avetisyan, M. Babenko, and A. Y. Drozdov, ''Dynamic performance–Energy tradeoff consolidation with contention-aware resource provisioning in containerized clouds,'' *PLoS ONE*, vol. 17, no. 1, Jan. 2022, Art. no. e0261856.

[45] K. Camboim, J. Ferreira, J. Araujo, and F. Alencar, ''Sustainability analysis in data center dense architectures,'' in *Proc. IEEE 9th Int. Conf. Cloud Netw. (CloudNet)*, Nov. 2020, pp. 1–6, doi: DOI: 10.1109/CloudNet51028.2020.9335791.

[46] G. Callou, P. Maciel, E. Tavares, E. Sousa, B. Silva, J. Figueiredo, C. Araujo, F. S. Magnani, and F. Neves, ''Sustainability and dependability evaluation on data center architectures,'' in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2011, pp. 398–403, doi: 10.1109/ICSMC.2011.6083698.

[47] D. Rodríguez-Gracia, J. A. Piedra-Fernández, L. Iribarne, J. Criado, R. Ayala, J. Alonso-Montesinos, and C.-U. M. D. Las Mercedes, ''Microservices and machine learning algorithms for adaptive green buildings,'' *Sustainability*, vol. 11, no. 16, p. 4320, Aug. 2019, doi: 10.3390/su11164320.

[48] A. Singh, G. S. Aujla, and R. S. Bali, ''Container-based load balancing for energy efficiency in software-defined edge computing environment,'' *Sustain. Comput., Informat. Syst.*, vol. 30, Jun. 2021, Art. no. 100463.

[49] A. Al-Moalmi, J. Luo, A. Salah, K. Li, and L. Yin, ''A whale optimization system for energy-efficient container placement in data centers,'' *Exp. Syst. Appl.*, vol. 164, Feb. 2021, Art. no. 113719.

[50] M. Xu, A. N. Toosi, and R. Buyya, ''IBrownout: An integrated approach for managing energy and brownout in container-based clouds,'' *IEEE Trans. Sustain. Comput.*, vol. 4, no. 1, pp. 53–66, Jan. 2019, doi: 10.1109/TSUSC.2018.2808493.

[51] D. Nikhil, B. Dhanalaxmi, and K. S. Reddy, ''The evolution of cloud computing and its contribution with big data analytics,'' in *Innovative Data Communication Technologies and Application*. Springer, 2020, pp. 332–341.

[52] J. Varia. (2008). *Cloud Architectures*. [Online]. Available: https://WhitePaperAmazon,jineshvaria.s3.amazonaws.com/public/cloudarchitectures-varia.pdf

[53] A. Huth and J. Cebula, ''The basics of cloud computing,'' *United States Comput.*, pp. 1–4, Apr. 2011.

[54] K. Kaur, S. Garg, G. Kaddoum, F. Gagnon, and D. N. K. Jayakody, ''EnLoB: Energy and load balancing-driven container placement strategy for data centers,'' in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2019, pp. 1–6.

[55] Cloudflare. (2023). *What is Edge Computing?—Cloudflare*. [Online]. Available: https://www.cloudflare.com/pt-br/learning/serverless/glossary/what-is-edge-computing/

[56] R. Asif, K. Ghanem, and J. Irvine, ''Containerization: For over-the-air programming of field deployed Internet-of-Energy based on cost effective lpwan,'' in *Proc. 1st Int. Conf. Smart Syst. Emerg. Technol. (SMARTTECH)*, 2020, pp. 65–70.

[57] StackPath. *What is Fog Computing?—Stackpath*. [Online]. Available: https://www.stackpath.com/edge-academy/what-is-fog-computing/

[58] A. Katal, S. Dahiya, and T. Choudhury, ''Energy efficiency in cloud computing data centers: A survey on software technologies,'' *Cluster Comput.*, vol. 26, no. 3, pp. 1845–1875, Jun. 2023.

[59] S. S. Panwar, M. M. S. Rauthan, and V. Barthwal, ''A systematic review on effective energy utilization management strategies in cloud data centers,'' *J. Cloud Comput.*, vol. 11, no. 1, p. 95, Dec. 2022, doi: 10.1186/s13677-022-00368-5.

[60] M. Pesce. (2021). *Cloud Computing's Coming Energy Crisis the Cloud's Electricity Needs are Growing Unsustainably*. [Online]. Available: https://spectrum.ieee.org/cloud-computings-coming-energy-crisis

[61] A. Kozakiewicz and A. Lis, ''Energy efficiency in cloud computing: Exploring the intellectual structure of the research field and its research fronts with direct citation analysis,'' *Energies*, vol. 14, no. 21, p. 7036, Oct. 2021. doi: 10.3390/en14217036.

[62] R. Zhang, Y. Chen, B. Dong, F. Tian, and Q. Zheng, ''A genetic algorithm-based energy-efficient container placement strategy in CaaS,'' *IEEE Access*, vol. 7, pp. 121360–121373, 2019.

[63] J. Luo, L. Yin, J. Hu, C. Wang, X. Liu, X. Fan, and H. Luo, ''Container-based fog computing architecture and energy-balancing scheduling algorithm for energy IoT,'' *Future Gener. Comput. Syst.*, vol. 97, pp. 50–60, Aug. 2019, doi: 10.1016/j.future.2018.12.063.

[64] S. F. Piraghaj, A. V. Dastjerdi, R. N. Calheiros, and R. Buyya, ''A framework and algorithm for energy efficient container consolidation in cloud data centers,'' in *Proc. IEEE Int. Conf. Data Sci. Data Intensive Syst.*, Dec. 2015, pp. 368–375, doi: 10.1109/DSDIS.2015.67.

[65] R. Canosa, A. Tchernykh, J. M. Cortés-Mendoza, R. Rivera-Rodriguez, J. L. Rizk, A. Avetisyan, Z. Du, G. Radchenko, and E. R. Concepción Morales, ''Energy consumption and quality of service optimization in containerized cloud computing,'' in *Proc. Ivannikov Ispras Open Conf. (ISPRAS)*, 2018, pp. 47–55, doi: 10.1109/ISPRAS.2018.00014.

[66] S. Mendes, J. Simao, and L. Veiga, ''Oversubscribing micro-clouds with energy-aware containers scheduling,'' in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, 2019, pp. 130–137.

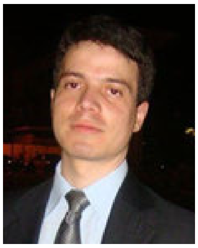
GABRIEL ARAÚJO received the degree in systems analysis and development from the Federal Institute of Piauí, in 2019, where he is currently pursuing the master's degree in computer science. He is also a member of the Applied Research Group for Distributed Systems (PASID). His research interests include distributed distribution systems, with an emphasis on container migration and performance evaluation techniques.

VANDIRLEYA BARBOSA received the degree in systems analysis and development from the Federal Institute of Piauí, in 2019, where she is currently pursuing the Graduate degree in information systems. She is also a member of the Applied Research Group for Distributed Systems (PASID). Her research interests include distributed distribution systems, with an emphasis on container migration and performance evaluation techniques.

LUIZ NELSON LIMA received the degree in systems analysis and development from the Federal Institute of Piauí, in 2019, where he is currently pursuing the Graduate degree in information systems. He is also a member of the Applied Research Group for Distributed Systems (PASID). His research interests include distributed distribution systems, with an emphasis on container migration and performance evaluation techniques.

ARTHUR SABINO received the degree in systems analysis and development from the Federal Institute of Piauí, in 2019, where he is currently pursuing the Graduate degree in information systems. He is also a member of the Applied Research Group for Distributed Systems (PASID). His research interests include distributed distribution systems, with an emphasis on container migration and performance evaluation techniques.

CARLOS BRITO received the degree in systems analysis and development from the Federal Institute of Piauí, in 2019, where he is currently pursuing the master's degree in computer science. He is also a member of the Applied Research Group for Distributed Systems (PASID). His research interest includes distributed distribution systems, with an emphasis on performance evaluation techniques.



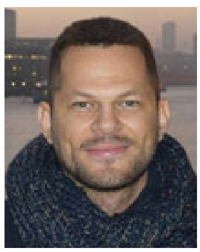
IURE FÉ received the degree in computer science from the Federal University of Piauí and the master's degree in computer science from the Federal University of Pernambuco. He is currently a Systems Analyst, who works for Brazilian Army. He also conducts research in the area of distributed systems and performance evaluation of systems.

PAULO REGO received the bachelor's, master's, and Ph.D. degrees in computer science from the Federal University of Ceará, in 2009, 2012, and 2016, respectively. He is currently an Adjunct Professor with the Science Center, Department of Computing, Federal University of Ceará (UFC). In 2015, he received a Sandwich Doctorate Scholarship from the Science Without Borders Program, Cisco Systems Inc., USA. He works in the area of computer science, with an emphasis on distributed systems and computer networks. His main research interests include mobile and cloud computing, microservices, OpenTelemetry, the Internet of Things, and software defined networks.

EUNMI CHOI received the B.S. degree in computer science from Korea University, in 1988, and the M.S. and Ph.D. degrees in computer science from Michigan State University, USA, in 1991 and 1997, respectively. She is currently a Professor with Kookmin University, South Korea. Her current research interests include big data infra system and analysis, cloud computing, intelligent systems, information security, parallel and distributed systems, and SW architecture and modeling. Before joining Kookmin University, in 2004, she has been an Assistant Professor with Handong University, South Korea, since March 1998. She is currently the Head of the Distributed Information System and Cloud Computing Laboratory, Kookmin University. She received the Top Student Award for the B.S. degree.

DUGKI MIN received the B.S. degree in industrial engineering from Korea University, in 1986, and the M.S. and Ph.D. degrees in computer science from Michigan State University, in 19991 and 1995, respectively. He is currently a Professor with the Department of Computer Science and Engineering, Konkuk University. His research interests include cloud computing, distributed and parallel processing, big data processing, intelligent processing, software architecture, and modeling and simulation.

TUAN ANH NGUYEN (Member, IEEE) received the M.Sc. and B.Eng. degrees in mechatronics from Hanoi University of Science and Technology (HUST), in 2010 and 2008, respectively, and the Ph.D. degree in computer science and systems engineering from the Department of Computer Engineering, Korea Aerospace University, in 2015. He currently holds the position of an Academic Research (Assistant) Professor in aerospace design-airworthiness research with the Konkuk Aerospace Design-Airworthiness Research Institute (KADA), Konkuk University, Seoul, South Korea. He has been the Technical Project Leader of the Project ''Artificial Intelligence Digital Twin Research for Urban Air Mobility (UAM)'' funded by Korea National Research Foundation (NRF) (2020–2029). He is currently a Professional Member of AIAA, IEEE Computer, Robotics and Automation (IEEE RAS), Aerospace and Electronic Systems (IEEE AESS), and Reliability Societies. Prior to his current role, he was a Research (Assistant) Professor with the Office of Research, University-Industry Cooperation Foundation, Konkuk University, from March 2016 to February 2020, where he also held the position of a Postdoctoral Research Associate with the Distributed Multimedia Systems Laboratory (DMS Laboratory), from August 2015 to February 2016, and a Ph.D. Research Associate with the Network Security and Systems Laboratory (NS Laboratory), Korea Aerospace University, from 2011 to 2015. Additionally, he was a Research Engineer with the TRON Laboratory and F-Space Laboratory, FPT Software and FPT Technology Research Institute (FTRI), Hanoi, Vietnam, in 2008 and (2009–2010), respectively. His current research interests include dynamics and control theory and systems, AI-based digital twin systems and methods, computer science, and software engineering, with a specialization in dependable, autonomous, intelligent systems, dependable computing and fault-tolerance of systems and networks, and mechatronics and aerospace robotic systems.



FRANCISCO AIRTON SILVA received the B.S. degree in information systems from Instituto Federal do Piauí (IFPI), in 2008, and the M.S. and Ph.D. degrees in computer science from Universidade Federal de Pernambuco (UFPE), in 2013 and 2017, respectively. He is currently a Professor in information systems with IFPI. In 2015, he studied with Sapienza Università di Roma. He was a Collaborative Researcher in multinational technology companies, such as EMC and OKI. His research interests include distributed systems, with an emphasis on cloud-fog-edge integration, mobile computing, and systems performance evaluation techniques.