

Review

Edge AI for SD-IoT: A Systematic Review on Scalability and Latency

Ernando P. Batista, Jr. ^{1,2} , Alex Santos ², Maycon Peixoto ¹ , Gustavo Figueiredo ¹  and Cassio Prazeres ^{1,*} 

¹ Institute of Computing, Federal University of Bahia (UFBA), Salvador 40170-110, BA, Brazil; ernando.passos@ifba.edu.br (E.P.B.J.); maycon.leone@ufba.br (M.P.); gustavobf@ufba.br (G.F.)

² Computer Science Department, Federal Institute of Bahia (IFBA), Salvador 40301-015, BA, Brazil; alex.silva@ifba.edu.br

* Correspondence: prazeres@ufba.br

Abstract

The growing demand for IoT applications in highly dynamic environments with multiple connected devices introduces significant scalability and low-latency challenges. In the context of software-defined networking (SDN) integrated with Edge environments, adopting machine learning (ML) techniques has emerged as a promising approach to meet these requirements. This study presents a Systematic Literature Review (SLR) that identifies and analyzes ML-based solutions applied to Software-Defined Internet of Things (SD-IoT) infrastructures in Edge environments, emphasizing improving low latency and scalability. Following established methodological best practices, we conducted the review, including a clear definition of research questions, well-defined inclusion and exclusion criteria, a structured search protocol, and multiple scientific databases. Based on the analysis of selected studies, the main strategies employed to enhance network performance are categorized, along with the level of fidelity and complexity of the experimental environments used, and the realism and applicability of the proposed solutions are discussed. Furthermore, drawing from the context of the selected studies, the most recurrent ML approaches are presented—including supervised, unsupervised, and reinforcement learning methods—along with a discussion of their advantages and limitations in dynamic network scenarios. By compiling and organizing the contributions from the literature, this paper provides a comprehensive overview of the state of the art in applying ML to SD-IoT networks, shedding light on current trends, existing gaps, and research opportunities aimed at building more intelligent and adaptable solutions for IoT environments.

Keywords: internet of things; machine learning; edge AI; software-defined networking



Academic Editor: Grigorios E. Koulouras

Received: 27 October 2025

Revised: 16 December 2025

Accepted: 19 December 2025

Published: 27 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The implementation of Internet of Things (IoT) environments and applications encompasses a wide range of possibilities, and the continuous growth in the number of connected devices imposes increasing demands on resources for their efficient management. In this context, implementing IoT infrastructures capable of providing a high level of scalability and low latency is one of the current challenges in the area [1,2]. These requirements are met by the Edge Computing paradigm by providing IoT management capabilities closer to the devices, leading to shorter response times and faster message exchanges. However, when compared to Cloud Computing, Edge has limited capacity to adapt to fluctuations in processing demands from IoT devices due to its lower processing capacity [3].

Scalability is a fundamental system characteristic that is highly related to the infrastructure's ability to maintain full operation despite the increase in demand for services and processing. In IoT deployments, it is an imperative requirement due to the high degree of dynamism of IoT applications and devices. Studies on scalability in IoT environments have explored a variety of key challenges, including the prediction of critical requests [4], Load Balancing [5], data/Flow Management in publication/subscription protocols [6], and the analysis and classification of service interactions [7].

SDN contributes to these goals by providing centralized, programmable, and adaptive network management. SDN facilitates dynamic routing and real-time traffic control, which supports both low-latency communication and the scalable operation of large IoT environments [8]. Even so, SDN alone is often insufficient to handle the high variability of IoT traffic.

Additionally, the growing complexity and demand for higher levels of scalability and low latency in IoT applications have led to the emergence of intelligent solutions based on ML. By incorporating predictive capabilities, these approaches enhance system adaptability and improve resource utilization in Edge Computing [8]. As a result, the convergence of IoT, Edge Computing, and ML has become central to implementing modern intelligent IoT systems, with the potential to drive advancements in both theory and practical applications.

Given the relevance of these requirements, this SLR investigates ML-based approaches designed to address scalability and low-latency challenges in SD-IoT environments. The objective is to highlight potential strategies, scenarios, and gaps that drive ongoing studies and results. In summary, the contributions of this paper include the following:

- Identify and categorize the main strategies employed in SD-IoT solutions to address scalability and low-latency challenges;
- Provide a comprehensive overview of the current state of research on SD-IoT infrastructures, with a specific focus on scalability and low latency in Edge environments;
- Present detailed results and analyses in response to the four research questions guiding this review, offering insights to inform and support future investigations;
- Provide the reference list for the 25 primary studies, establishing a comprehensive knowledge base for future researchers delving into the field of this SLR;
- Highlight existing research gaps and open challenges for future work to advance the development of intelligent, scalable, and low-latency SD-IoT solutions.

The remainder of this paper is organized as follows: Section 3 provides an overview of the SLR and outlines the planning stage of the SLR, including the formulation of research questions and the identification of related reviews. Section 4 details the conducting stage, encompassing the search strategies and study selection criteria. Sections 5 and 6 summarize the selected studies and discuss the review findings about the research questions. Section 10 examines the gaps and challenges identified in the literature. Section 11 addresses potential threats to the validity of the SLR. Finally, Section 12 presents the concluding remarks.

2. Context Formulation

In SD-IoT environments integrated with Edge Computing, the proliferation of heterogeneous devices and dynamic communication demands introduces two critical and interconnected challenges: scalability and low latency. These challenges become particularly complex when systems aim to support real-time or near-real-time operations while maintaining the capacity to adapt to growing workloads and varying network conditions.

To address these issues, in this work, we adopt a conceptual model in which the SD-IoT infrastructure is organized into three core logical layers: (i) SD-IoT Devices, (ii) SD-Control, and (iii) SD-IoT Applications.

- SD-IoT Devices (D): IoT nodes (sensors and IoT gateways) responsible for sensing, actuating, transmitting data, and local processing;
- SD-Control: Centralized or distributed software-defined controllers managing data flow, routing, and Resource Allocation;
- SD-IoT Applications: Application-level services, orchestration mechanisms, and intelligent management functions that rely on the SD-Control layer support.

For example, in the SD-IoT framework presented by [9], the architecture explicitly adopts a three-layer model (Application, Control, Infrastructure), where IoT devices and gateways form the infrastructure layer, SDN controllers compose the control layer, and applications operate in the upper layer. Similarly, Siddiqui et al. [10] provide a generalized SDN-IoT reference architecture that also relies on three logical planes: Data Plane, Control Plane, and Application Plane, with IoT nodes positioned in the data/infrastructure plane, controllers in the control plane, and management or intelligence functions in the application plane. Our three-layer abstraction (SD-IoT Devices, SD-Control, SD-IoT Applications) corresponds directly to these models.

To formalize the scalability–latency aspects studied in this review, let

- $D = d_1, d_2, \dots, d_n$ be the set of SD-IoT sensors devices;
- $G = g_1, g_2, \dots, g_m$ be the set of SD-IoT gateways providing distributed processing capabilities;
- $C = c_1, c_2, \dots, c_k$ be the set of SD-Control controllers.

We define the following system functions relevant to scalability and latency:

- $\tau(d_i, e_j)$: latency of communication and processing between SD-IoT sensor device d_i and SD-IoT gateway g_j ;
- $\sigma(g_j)$: processing load or scalability metric at SD-IoT gateway g_j ;
- $\rho(g_j, c_l)$: responsiveness of controller c_l in orchestrating services mediated through gateway g_j ;
- $\lambda(t)$: system load variation over time t , influencing both latency and scalability.

Given these definitions, the review focuses on three major problems extracted from the literature:

Problem 1: Minimizing End-to-End Latency

Identify techniques and architectures that minimize

$$\min \tau(d_i, g_j) + \rho(g_j, c_l)$$

while satisfying application-specific latency constraints.

Problem 2: Improving Scalability under Dynamic Loads

Maximize the number of concurrently supported devices while maintaining system stability:

$$\max |D| \quad \text{s.t.} \quad \sigma(g_j) < \sigma_{\max} \quad \forall g_j \in G$$

Problem 3: Intelligent Adaptation via Machine Learning

Leverage ML-based approaches (supervised, unsupervised, reinforcement) to dynamically adjust routing, Resource Allocation, and flow control. Define a learning function:

$$f_{\theta} : \mathcal{S} \rightarrow \mathcal{A}$$

where $\mathcal{S}$ is the state space (e.g., load, latency metrics, topology), and $\mathcal{A}$ is the action space (e.g., routing decision, resource adjustment), learned by minimizing a task-specific loss (e.g., latency, packet loss, energy usage).

These problems include mechanisms such as Routing Optimization, Flow Management, Resource Allocation, Load Balancing, and Service Provisioning using Edge AI models. In addition, synthesize the challenges addressed in the selected studies and provide the foundation for organizing strategies, solutions, and open research gaps in the context of Edge AI for SD-IoT.

The goal of this section is not to propose a new theoretical model or to define the exact scenarios addressed in the studies selected for the SLR. Instead, this section establishes a clear and unified conceptual basis that supports the reader in understanding the main challenges discussed throughout the paper. Given the diversity of architectures, evaluation metrics, and experimental conditions found in SD-IoT research, a common vocabulary, comprising logical layers, variables, and high-level optimization goals, helps to situate and ensure coherence in later comparisons.

In addition, the formulations introduced here should, therefore, be viewed as a general analytical guide. They do not influence which studies are selected for the SLR and do not restrict the characteristics of the solutions found in the literature. Each study is independently analyzed during the SLR process, and its specific details, architectures, and scenarios are examined in the data extraction and synthesis steps described in Sections 3 and 5.

3. The Review Methodology and Planning

This section describes the methodological foundation and planning of this Systematic Literature Review. It presents the approach adopted to define the research questions, the scope of investigation, and the criteria used for selecting and evaluating studies. The section also outlines the rationale for conducting the review, the adoption of the Population, Intervention, Comparison, Outcome, Context (PICOC) model, and the identification of related works that support the formulation of the research objectives. The following subsections detail the research questions, planning steps, and criteria that guided the subsequent stages of study selection and analysis.

3.1. Methodology

An SLR is a method used to identify, evaluate, and interpret data from relevant and available research on a specific topic of study [11]. The structuring of an SLR aims to promote a more comprehensive understanding, generate insights, and reveal potential areas for exploration within the topics studied.

We conduct the SLR following the studies and guidelines proposed by [11,12]. The SLR is structured into three main stages, each comprising sub-stages (see Figure 1). To enhance replicability and methodological transparency, we explicitly describe each stage of our SLR using a sequence of reproducible steps. These steps are:

- Define research questions using the PICOC model (Section 3.2.1);
- Perform initial identification of related works through exploratory analysis and snowballing (Section 3.2.2);
- Identify keywords and construct complete search strings (Section 4);
- Validate search strings internally with two reviewers (Section 4);
- Execute all search strings in selected databases (Section 4);
- Remove duplicate studies by verification (Section 4 and Section Data Extraction);
- Screen titles and abstracts using EC1-EC2 (Section 4);
- Conduct full-text screening using EC3-EC4 (Section 4);
- Apply the predefined quality assessment criteria (Section Data Extraction);
- Consolidate results and classify studies according to the SRQs (Sections 5 and 6).

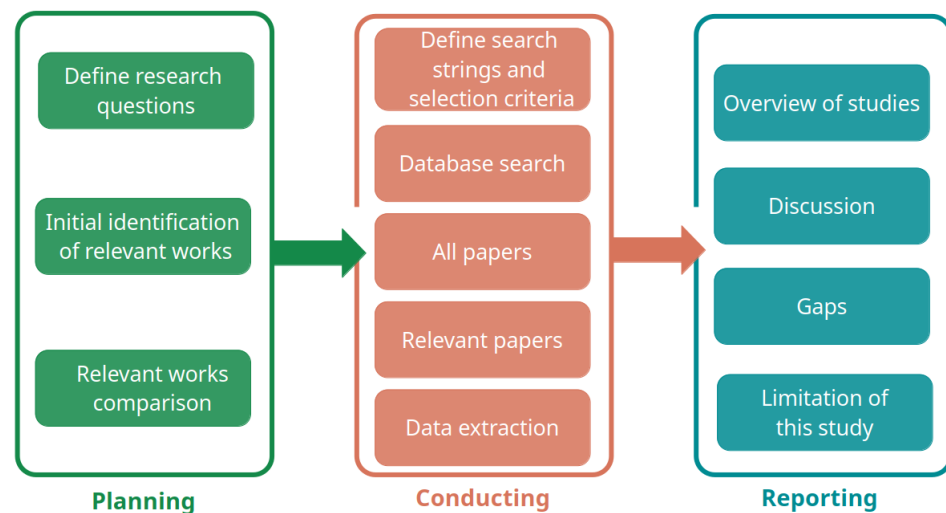


Figure 1. SLR Overview.

3.2. Planning

In this section, we outline the Planning of an SLR, an important step that establishes a comprehensive investigation of existing research. The Planning phase involves formulating research questions, defining the scope, and establishing inclusion and exclusion criteria to guide the review process. It also encompasses the development of a protocol that specifies the methodology for searching, selecting, and analyzing relevant studies [11]. Thus, in this section, we describe the planning sub-steps, as shown in Figure 1, which will support the subsequent conducting and reporting steps.

3.2.1. Research Questions

This section defines the research questions that will guide the SLR construction. The PICOC model is employed to support the formulation and development of these questions. This model defines criteria that support framing research questions [13]. The attributes are as follows: (i) population—the group of software users/testers, the research data collection site, or a group of industries/IT companies; (ii) intervention—the software methodology/tool/technology/procedure addressing a specific problem; (iii) comparison—the methodology/tool/technology/procedure with which the solution is being compared; (iv) outcomes—key factors obtained through the intervention; (v) context—the software context referenced in the previous items.

Table 1 presents the definition of the PICOC criteria applied to this SLR. It is worth noting that the Comparison criterion does not apply to this work, as we treat the SLR as primarily focused on summarizing evidence, identifying gaps, and positioning the research area of this thesis. As mentioned in the introduction of this chapter, this study (SLR) aims to investigate the current state of intelligent solutions and strategies (based on ML) in SD-IoT environments, focusing on low latency and scalability in Edge Computing. Therefore, the Main Research Question (MRQ) of this SLR, which will be addressed in this work, is:

“Which machine learning-based solutions and strategies proposed by the scientific community, when applied to SDN networks and IoT/Edge environments, have demonstrated improvements in low latency and/or scalability requirements?”

In addition, we developed a set of secondary research questions (SRQs). The SRQs aim to further define the objectives of this SLR by providing additional information on the methodology, metrics, and types of solutions used in the studies reviewed. In this context, the following SRQs were defined:

SRQ-1: *What specific problems/focuses are addressed in these studies to tackle scalability and low latency issues?* This SRQ aims to understand the specific problems that have

been investigated and how they relate to the low latency and scalability requirements in IoT/Edge applications.

SRQ-2: *What scenarios are used as case studies or experimental setups in the analyzed works?* This SRQ aims to gather information about the main test scenarios employed in the studies and understand how these scenarios align with the proposed solutions, particularly regarding meeting low latency and scalability requirements.

SRQ-3: *What are the behaviors, fidelity, and level of complexity involved in implementing the proposed solutions in real-world environments?* This SRQ aims to obtain insight into the degree of fidelity of the solutions to real-world conditions. Additionally, it aims to collect information on the practicality of deploying the proposed approaches in real or production environments, taking into account aspects such as source code availability, libraries, and supporting technologies.

SRQ-4: *What machine learning methods and types of comparative analyses or results were used in the studies?* This SRQ aims to identify and classify the machine learning methods or algorithms applied, as well as the types of comparisons and result analyses presented in the studies.

Table 1. Research questions as structured by the PICOC criteria.

Population	Studies published in the scientific literature on developing ML-based solutions applied to SD-IoT environments in Edge Computing focusing on low latency and/or scalability.
Intervention	ML-based algorithms, techniques, and approaches that support the development of SD-IoT applications, with a focus on low latency and/or scalability.
Comparison	Not applicable
Outcome	Effectiveness of SD-IoT solutions focused on low latency and/or scalability in Edge Computing environments.
Context	SD-IoT, Machine Learning and Edge Computing

3.2.2. Initial Identifications of Relevant Works

Based on the methodologies proposed in [11,12], we developed an approach for identifying and initially collecting relevant literature for this SLR. In the first stage of the identification process, we conducted an exploratory analysis of previously published SLRs and surveys that intersect with the themes of this review—SD-IoT, Edge Computing, and ML. This exploratory scan (Step 1) offers domain context and supports the justification of our SLR focus, while the systematic selection of primary studies is performed in Step 2.

In the second step, we collected and identified studies based on predefined keywords, selection criteria, and metrics outlined in the planning phase. These studies were sourced from key databases in Computer Science and Information Technology. In the third stage, we consolidated the studies gathered in the previous steps, performed a final filter to remove duplicates and irrelevant papers, and selected studies that addressed the research questions outlined in Section 3.2.1.

Step 1: Analysis of existing studies

Study [14] reviews network softwarization and the integration of ML techniques into SDN, Network Function Virtualization (NFV), and IoT ecosystems. The authors report advances in Load Balancing, dynamic data aggregation, path optimization, traffic classification, and energy-aware mechanisms. The review also emphasizes the need for implementations with higher fidelity and more consistent evaluations, particularly in Edge contexts.

Focused specifically on routing, ref. [15] analyzes ML-based approaches for path optimization in SDN. The authors provide detailed taxonomies, comparison tables, and qualitative analyses of ML-based routing techniques, highlighting their implementation contexts,

advantages, limitations, and evaluation metrics. For future research, the authors include the need for standardized datasets, practical implementations, and reproducible experiments.

SLR [10] examines SDN-based IoT management frameworks addressing scalability, Load Balancing, and security. The study observes that many proposals lack implementation details or practical evaluations and highlights the limited presence of AI-driven approaches in the SD-IoT domain.

Complementing these perspectives, ref. [16] surveys AI techniques in Edge environments, pointing to the benefits of SDN programmability for real-time decisions and reduced latency. It also stresses the importance of simulation tools and ML frameworks in enabling rapid prototyping of Edge solutions.

Survey [3] investigates Edge–Cloud architectures for Artificial Intelligence of Things (AIoT) systems and underscores the role of deep learning in data analysis and decision-making. Challenges related to heterogeneity, interoperability, and latency-sensitive tasks in resource-constrained environments are also highlighted, reinforcing the relevance of SDN and NFV in orchestrating distributed systems.

The work in [17] explores ML in Software-Defined Wireless Sensor Networks, discussing energy efficiency, delay reduction, and security. The authors identify pressing needs for improved controller placement, Edge-node scalability, and diversified testbeds.

Deep Reinforcement Learning (DRL) in IoT-Edge and SDN/NFV environments is the focus of [18], which examines how DRL-based solutions can reduce latency, enhance scalability, and minimize communication overhead, positioning DRL as a promising avenue for future SD-IoT optimizations.

Other studies also contribute to the broader context. Review [19] analyzes the task offloading in IoT–Edge–Cloud ecosystems, highlighting the importance of SDN for dynamic service allocation, latency reduction, and scalability in multi-controller deployments. Meanwhile, ref. [20] discusses ML-driven task offloading strategies in IoT/Edge environments, detailing architectures and intelligent workload distribution techniques that support low-latency decision-making.

Collectively, the works examined in this initial stage (see Table 2) reinforce the emerging convergence of IoT, SDN, Edge Computing, and ML, while also revealing persistent challenges, such as limited scalability evaluations, reduced experimental realism, and the scarcity of solutions explicitly targeting both latency and scalability.

As a continuation of identifying relevant literature, we reviewed the titles, abstracts, and introductions of the 40 studies identified in this section. The selection criteria involved excluding studies that were not directly relevant to our investigation, i.e., those that could not address the research questions. One reviewer (Batista) conducted this screening process and then met with a second reviewer (Santos) to compare their findings. A third independent reviewer (Prazeres) discussed and resolved any disagreements. As a result, six studies were selected as potentially relevant based on the criteria outlined above.

Table 2. Comparison of initial related works.

Authors	Year	Methodology	Domain	Approach Solutions	Main Goal
Alonso, Prieto and Corchado [18]	2021	Review	IoT, SDN, ML, Edge	Management Edge-IoT	Explores applications of Deep Reinforcement Learning techniques for managing SDN in Edge-IoT environments
Lone and Sofi [20]	2023	Review	IoT, ML, Edge, Cloud	Task offloading	Presents a review of task offloading strategies in Edge-based IoT environments with ML techniques
Zormati, Lakhlef and Ouni [14]	2024	Systematic Review	IoT, SDN, ML	Network Softwarization	Explores the state of the art in SDN, and ML, as well as their integration into IoT scenarios
Siddiqui et al. [10]	2022	Systematic Review	IoT, SDN	IoT frameworks, scalability, Load Balancing	Explores SDN-based frameworks for IoT management, addressing topics such as scalability, Load Balancing, and security

Table 2. Cont.

Authors	Year	Methodology	Domain	Approach Solutions	Main Goal
Singh and Gill [16]	2023	Survey	IoT, ML, Edge	Edge AI	Discusses the advantages of applying AI in the context of Edge and Edge Computing
Chang et al. [3]	2021	Survey	IoT, ML, Edge, Cloud	architectures	Addresses Edge-cloud architectures for AIoT
Jurado-Lasso et al. [17]	2022	Survey	SDN, ML	ML for software-defined wireless sensor networks	Describe the current state of Software-Defined Wireless Sensor Networks and their integration with ML techniques
Rezaee et al. [19]	2024	Review	IoT, SDN, Edge, Cloud	Task offloading and Task Management	Review studies on task offloading in IoT-Edge-Cloud environments, focusing on SDN networks and applications
Amin et al. [15]	2021	Survey	SDN, ML	Routing optimization	Presents ML techniques applied to Routing Optimization in SDN environments
This study	2025	Systematic Review	IoT, SDN, ML, Edge	Low Latency and Scalability	Explores the state of the art of IoT, SDN, and ML at the Edge with a focus on low latency and scalability.

4. Research Conducting

This section details the execution phase of the SLR, which operationalizes the planning strategy described in the previous section. Here, we describe the search procedures, databases, and keywords used to identify relevant studies, as well as the criteria and steps (Steps 2 and 3 in Figure 2) applied to select and refine the final pool of primary publications. The objective of this phase is to ensure methodological rigor and transparency in retrieving and filtering the literature, thereby enabling an accurate and reproducible synthesis of existing research on ML-based solutions for scalability and low latency in SD-IoT environments.

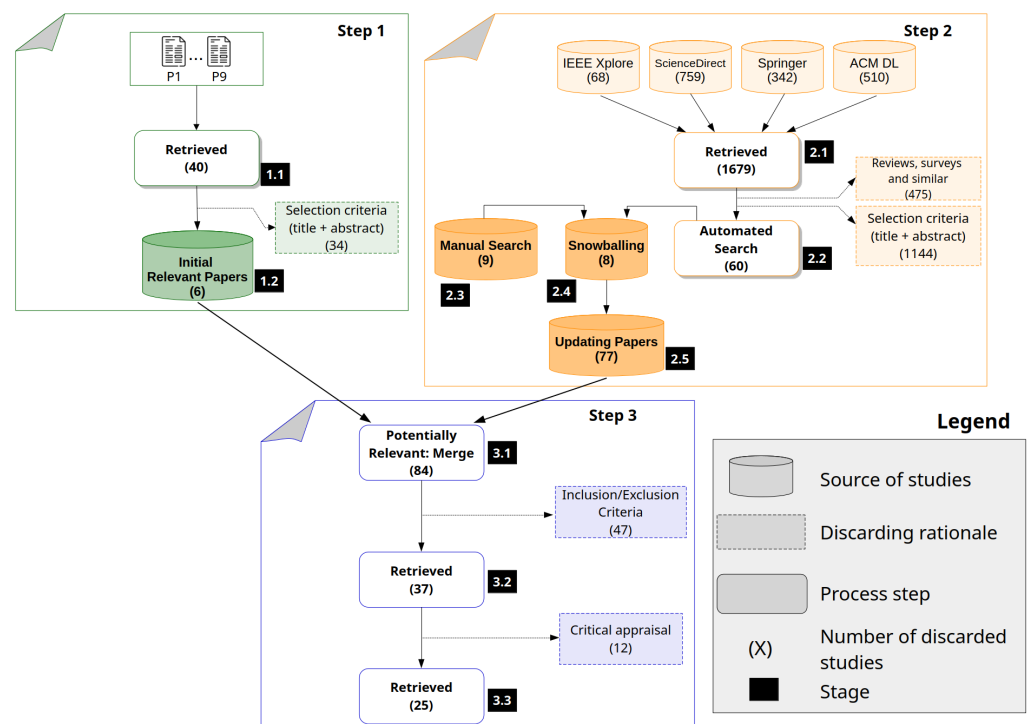


Figure 2. Study selection method. Based on [12].

Step 2: Gathering recent publications

The second step of the research process involved updating and expanding the list of primary studies. In this phase, we focused on literature published between 2020 and 2025 (until February). Searches were conducted in the following scientific databases: ACM Digital Library, IEEE Xplore Digital Library, ScienceDirect, and Springer (Table 3). These digital libraries were selected based on their prominence in Computer Science [21].

These databases are particularly valuable as they index publications from IEEE, ACM, Springer, and Elsevier, collectively covering many of the leading works in Computer Science. Consequently, they are expected to encompass significant papers relevant to the topics addressed in the MRQ and SRQs.

Based on the research questions and the studies analyzed in Step 1, we identified a set of keywords to guide the search process. Variations of the terms “*internet of things*”, “*software defined networking*”, “*machine learning*”, “*fog computing*”, “*edge computing*”, “*low latency*”, and “*scalability*” were used to construct the search queries. These queries were adapted to the syntax requirements and capabilities of the search engines for each data source. Table 4 presents the specific search strings applied in each search engine and the number of results retrieved from each.

Table 3. Research Databases.

Engine	URL
ACM Digital Library	https://dl.acm.org/ (Accessed on 26 January 2025)
IEEE Xplore	https://ieeexplore.ieee.org/ (Accessed on 28 January 2025)
Science Direct	https://sciencedirect.com/ (Accessed on 24 January 2025)
Springer	https://www.springer.com/ (Accessed on 26 January 2025)

Table 4. Detailed search strings applied in the automated search engines. Execution Date for All Searches: January 2025.

Engine	Search String	Results
ACM Digital Library	(“Software Defined Networking” OR “SDN”) AND (“Internet of Things” OR “IoT”) AND (“Machine Learning”) AND (“Latency” OR “Scalability”)	510
IEEE Xplore	(Software Defined Networking OR SDN) AND (Internet of Things OR IoT) AND (Machine Learning) AND (Latency OR Scalability) AND (Fog Computing OR Edge Computing)	68
Science Direct	(“SDN” OR “Software Defined Networking”) AND (“Machine Learning”) AND (“IoT” OR “Internet of Things”) AND (“Latency” OR “scalability”) AND (“Fog” OR “Edge”) All Sources (Computer Science)	759
Springer	(‘SDN’) AND (‘Machine Learning’) AND (‘IoT’)	342

During the search and collection phase across the selected databases, we retrieved a total of 1679 publications, as detailed in Stage 2.1 (see Figure 2). This initial set included related works, conference proceedings, literature reviews, editorials, and similar studies. We excluded 475 of these publications, as they were already considered under different criteria within this SLR scope. The next activity involved reading the title and abstract of each remaining paper, following the same procedure as in the previous phase. In this step, 1144 publications were identified as irrelevant to the scope of this SLR, as they did not align with the themes defined by the main research question. The exclusion criteria applied to the publications between stages 2.1 and 2.2 are defined in items EC1 and EC2 of Table 5. As a result of this filtering process, a total of 60 publications were retained from the automated search, as illustrated in stage 2.2 of Figure 2.

Following the automated search, we conducted a manual search to enhance coverage [22]. We specifically searched key journals that were not indexed in the earlier phases. Table 6 lists the venues (search locations) and their standard abbreviations. This manual search resulted in retrieving an additional nine studies, as shown in step (2.3) of Figure 2.

Table 5. Exclusion criteria.

Exclusion Criteria		Rationale
EC1	Abstracts	We excluded prefaces, editorials, reviews, and summaries of tutorials, panels, and poster sessions.
EC2	Research paper out of scope	By analyzing the abstract section of a study, it is possible to figure out what the topic under investigation is about. The study should present initiatives related to the topic of the research strings.
EC3	Duplicated and Extended studies	The study must be unique. When duplicated papers of a study exist in different versions that appear as books, journal papers, conference and workshop papers, we include only the most complete version of the study and exclude the others.
EC4	Research paper metrics	When analyzing the abstract, introduction, and results, it is verified whether the metrics evaluated in the paper are in accordance with the MRQ topics.

Table 6. Venues subject to manual search.

Journals
Internet of Things Journal—Elsevier
Smart Cities Journal—MDPI
Applied Sciences—MDPI
Sensors—MDPI
Computer Networks—Elsevier
Conferences
International Conference on the Internet of Things
International Conference on Information Networking

In addition, we applied the process called snowballing. This process refers to using the reference list of a paper or the citations to the paper to identify additional papers [23]. From this task, we obtained eight studies, as Figure 2 illustrates in Step 2.4. Thus, at the end of Step 2, we revealed a set of 77 new publications.

Step 3: Primary study selection strategy

By merging the results from Steps 1 and 2, the list of potentially relevant studies was then composed of 84 publications. Item (3.1) in Figure 2 shows such an amount.

Between phases 3.1 and 3.2, we applied the exclusion criteria EC3 and EC4, as defined in Table 5. The primary objective of this filtering step was to eliminate studies that, although aligned with the overarching themes of the MRQs, did not address or evaluate the specific metrics outlined in the MRQs—namely, low latency and scalability. In addition, we removed duplicate studies. As a result of this screening process (phase 3.2), we obtained a final pool of 37 studies for full-text review.

Data Extraction

After using the criteria to select relevant papers (step 3.2), a quality assessment was performed on the remaining papers. We conducted a comprehensive analysis of the 37 filtered studies to collect the data necessary to answer the research questions and to measure their quality.

The quality assessed in each study obtained in step 3.2 was based on a set of quality criteria. We extracted the criteria mainly from the questionnaire for quality assessment proposed by [24]. Based on the questionnaire used to critically evaluate the quality of the selected studies, we developed seven closed-ended questions, as listed in Table 7. Taken together, the criteria provided a measure of the extent to which we could be confident that findings of a particular study could provide the review with a valuable contribution. The criteria covered the four main issues pertaining to quality that need to be considered when appraising the studies identified in the review:

- Reporting: Reporting of the study's rationale, aims, and context.
- Rigor: Has a thorough and appropriate approach been applied to key research methods in the study?
- Credibility: Are the findings well-presented and meaningful?
- Relevance: How useful are the findings to the software industry and the research community?

Each of the seven criteria was answered with either “yes” (1), “partially” (0.5), or “no” (0). Then, a quality assessment score was given to a study by summing up the scores for all the questions in the study. The resulting total quality score for each study ranged from 0 (very poor) to 7 (very good).

Table 7. Quality assessment questions

Question	Issue
Is there a clear statement of the aims of the research?	Reporting
Is there an adequate description of the context/scenarios/setup in which the research was carried out?	Reporting
Is the study valuable for research or practice?	Relevance
Was the modeling and setup of the experiments sufficiently rigorous?	Rigor
Were the experimental results in a way that addressed the research issue?	Rigor
Is there a clear statement of findings?	Credibility
Were the metrics clearly analyzed in the results described?	Credibility

We used quality assessment criteria for both synthesis purposes and final filtering of papers. Thus, the threshold for exclusion was studies that did not meet any of the seven criteria or obtained more than two “partial” values. Upon completion of the quality evaluation assessment, the set of selected primary studies was composed of 25 studies, illustrated as stage (3.3) and listed in Table 8. One reviewer (Batista) carried out the reading and completion of the extraction form for each paper. A second reviewer (Prazeres) resolved any discrepancies or errors identified during this process.

Table 8. Summary of selected works.

ID	Title	Main Contribution	Year
[25]	Dynamic Fog Intelligence with Flow Control for Green Internet of Things	Application-aware and delay-sensitive service provisioning in SDN-IoT	2022
[26]	Collaborative Flow-Identification Mechanism for Software-Defined Internet of Things	Traffic identification mechanism for secure SDN-IoT	2022
[27]	Asynchronous Federated Learning for Resource Allocation in Software-Defined Internet of UAVs	Resource Allocation algorithm integrating SDN-IoT UAV	2024
[28]	SDNQ: A Novel SDN Controller for the Management of IoT Video Flows in the Edge/Cloud Continuum	Managing IoT video flow	2024
[29]	Flow-level Dynamic Bandwidth Allocation in SDN-enabled Edge Cloud using Heuristic Reinforcement Learning	Flow-level dynamic bandwidth allocation	2021
[30]	Predictive mobility and cost-aware flow placement in SDN-based IoT networks: a Q-learning approach	Efficient flow entry placement	2024

Table 8. *Cont.*

ID	Title	Main Contribution	Year
[31]	A QoE-driven SDN traffic management for IoT-enabled surveillance systems using deep learning based on Edge Cloud Computing	Predict the optimal routes by minimum delay	2023
[32]	Improved Exploration Strategy for Q-Learning Based Multipath Routing in SDN Networks	Congestion avoidance for multipath flow routing	2024
[33]	Efficient deep-reinforcement learning aware Resource Allocation in SDN-enabled fog paradigm	Resource Allocation that handles offload workloads	2022
[34]	Triple Tier Framework for Intellectual Edge Assisted Multicontroller Load Balancing in SDN	Sensitivity-aware flow classification and Controller Load Balancing	2024
[35]	Routing Optimization Based on Distributed Intelligent Network Softwarization for the Internet of Things	Optimize routing in softwarized IoT networks using Federated Deep Reinforcement Learning	2025
[36]	A Framework for In-network Inference using P4	Managing flow tables from ML inference and traffic classification	2024
[37]	Machine learning enabled network and task management in SDN-based Fog architecture	Intelligent link weight adjusts to find the optimal path and allocation	2023
[38]	Deep Reinforcement Learning for energy-aware task offloading in joint SDN-Blockchain 5G massive IoT Edge network	Blockchain-based Deep Reinforcement Learning approach to enable energy-aware task scheduling and offloading	2022
[39]	Reliable paths prediction with intelligent data plane monitoring enabled reinforcement learning in SD-IoT	Intelligent network monitoring, reliability, and routing in SDN-IoT	2024
[40]	A Deep Reinforcement Learning-based multi-optimality routing scheme for dynamic IoT networks	Reinforcement Learning-based Routing schemes combined with Multi-optimality routing criteria	2021
[41]	Smart SDN Management of Fog Services	Adapt to significant changes in system load and preserve the users' QoS	2020
[42]	SDN-Enabled Adaptive and Reliable Communication in IoT-Fog Environment Using Machine Learning and Multiobjective Optimization	Reliability of links in a multihop and SDN-enabled IoT Fog	2021
[43]	Intelligent Resource Management at the Edge for Ubiquitous IoT: An SDN-Based Federated Learning Approach	Ensuring high QoS in a ubiquitous SDN-IoT network by cache management	2021
[44]	An SDN-Enabled Framework for a Load-Balanced and QoS-Aware Internet of Underwater Things	Multicontroller Load-balanced and QoS-aware software defined Internet of UnderWater	2023
[45]	A QoS Optimization Technique with Deep Reinforcement Learning in SDN-Based IoT	Resource management and allocation in the control plane of SDN-IoT networks	2021
[46]	Energy-efficient clustering and routing algorithm for large-scale SDN-based IoT monitoring	Q-routing/Q-learning algorithm for intelligent management of the global network	2020
[47]	A New SDN-Based Routing Protocol for Improving Delay in Smart City Environments	New routing protocol based on SDN and Naive Bayes solution to improve the delay	2020
[48]	Reinforcement Learning based Transmission Range Control in Software-Defined Wireless Sensor Networks with Moving Sensor	Transmission range control is adjusted based on reinforcement learning	2020
[49]	A new intelligent scheduler to improve reactive OpenFlow communication in SDN-based IoT data streams	Management of flow tables from ML prediction of the IoT data streams	2024

5. Reporting

We used the extracted data to answer our research questions. In this section, we initially give an overview of the selected studies with respect to their publication venues. Then, we answer each research question based on the extracted data.

Characteristics of the Studies

Based on the 25 studies selected in this SLR (step 3.3), Figure 3 presents the temporal distribution of publications between 2020 and 2025. The year 2024 accounts for the highest number of studies, comprising 32.00% (8 studies) of the total. Both 2020 and 2022 contributed 16.00% (4 studies each), while 2021 represented 20.00% (5 studies). In contrast, 2023 contributed 12.00% (3 studies), and 2025 accounts for only 4.00% (1 study). It is important to note that the inclusion of studies from 2025 reflects those that were publicly available at the time of the search, conducted at the end of January.

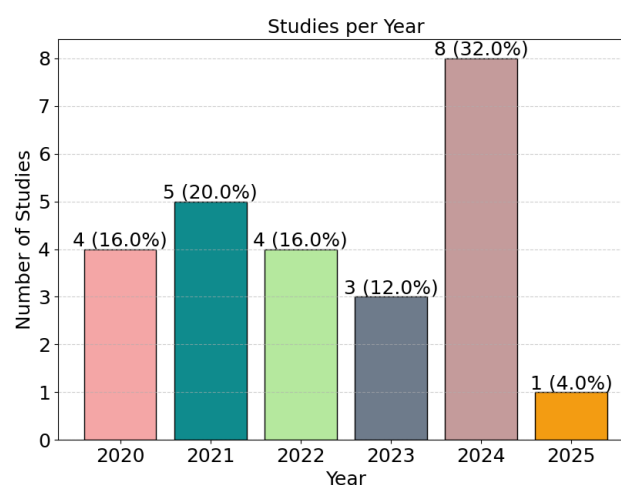


Figure 3. Temporal distribution of the 25 primary studies selected in the SLR, organized by publication year (2020–2025).

Considering the distribution of the selected studies according to their publication venues, among the 25 analyzed studies, 72.00% (18 studies) were published in journals, while the remaining 28.00% (7) appeared in conferences. This distribution suggests that most of the selected works are comprehensive scientific contributions, encompassing proposal, modeling, and experimental validation, given the typical rigor and completeness of journal publications. Notably, the IEEE Internet of Things Journal, a well-established venue in IoT solutions, accounts for the largest number of selected journal papers (five studies). The IEEE International Conference on Communications (ICC) stands out within the conference category, with two studies included in the selection.

In the database context, the IEEE Xplore database accounts for the most significant proportion, comprising 40.00% of the selected studies (10 studies), followed by the Springer database with 24.00% (6 studies). ACM DL and Science Direct each contribute 12.00% (three studies). Other datasets with manual searches contributed 12% (three studies).

6. Analysis and Discussion

In the previous section, we presented the temporal distribution of the selected studies by year, publication type, and database source. This section addresses the research questions by analyzing and categorizing the selected studies. In addition, the analyses presented in this section build on the methodological process described in Section 3 and use the basic concepts introduced in Section 2 to keep the discussion consistent. These concepts act only

as a common reference when reporting how each study addresses latency, scalability, and SD-IoT components.

In this context, Table 8 summarizes the authors, titles, publication years, and main contributions of each of the 25 studies included in our systematic literature review, labeled [25–49]. These studies illustrate the breadth and evolution of research in the field, providing valuable insights into the current landscape of the main research question and its related subtopics. Consequently, the analyses and discussions presented in this section are guided by the main and secondary research questions.

6.1. Strategies to Handle Low Latency and Scalability Issues (SRQ-1)

This subsection addresses research question SRQ-1 by examining the strategies employed in ML-based applications within IoT, Edge, and Fog Computing environments, particularly concerning low latency and scalability requirements. Based on the analysis of the selected studies, these strategies were categorized into five groups: (i) Load Balancing; (ii) Routing Optimization; (iii) Resource Allocation; (iv) Flow Management; and (v) Service Provisioning. Figure 4 illustrates the overall distribution of studies according to these categories. Considering that some studies present solutions that span multiple strategic categories, Routing Optimization presents the largest number of solutions (nine studies), followed by Flow Management (eight studies) and Resource Allocation (seven studies). Load Balancing (four studies) and Service Provisioning (two studies) finalize the distribution of the strategies used in the works selected in this SLR. In addition, the individualized categorization of these studies is in Table 9.

As illustrated in Figure 5, the adoption of different ML strategies has varied over time, with an increase in studies focused on Flow Management in 2024 and a consistent trend in Routing Optimization over the years. This trend reflects the growing importance of adaptive and efficient data flow mechanisms in response to the increasing complexity and dynamism of SD-IoT environments. Other strategies, such as Resource Allocation and Load Balancing, remain consistently relevant, though less dominant.

Table 9. Classification of the selected studies according to the main ML-based strategies adopted to address low-latency and scalability requirements in SD-IoT. This Table distinguishes whether each strategy explicitly targets low latency, scalability, or both, and lists the corresponding primary studies.

	Total	Low Latency	Total	Scalability
		Studies		Studies
Load Balancing	4	[31,33,34,44]	2	[34,44]
Routing Optimization	9	[31,32,35,37,39,40,42,46,47]	3	[35,40,47]
Resource Allocation	6	[27,33,37,38,45,48]	1	[43]
Flow Management	8	[26,28–30,34,36,37,49]	1	[34]
Service Provisioning	2	[25,41]	0	–

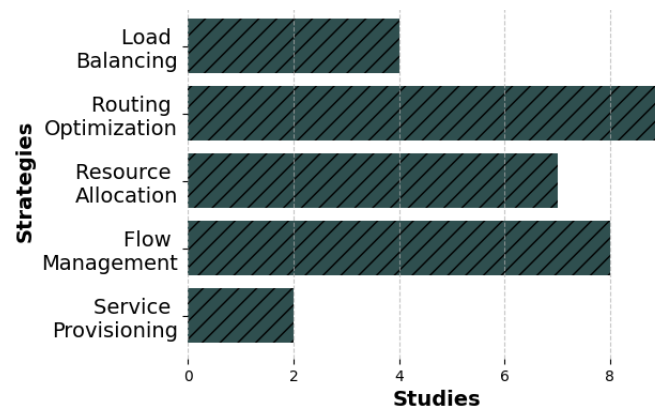


Figure 4. Distribution of the selected studies according to the main ML-based strategies adopted to address scalability and low-latency challenges in SD-IoT. A single study may contribute to multiple strategy categories.

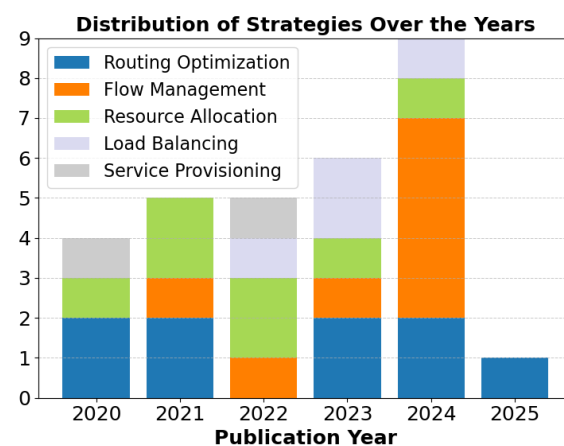


Figure 5. Annual evolution of the main ML-based strategies employed in the reviewed studies. The Figure illustrates how research focus has shifted over time.

Routing Optimization leverages centralized control and global network visibility provided by SDN to dynamically adjust routing paths based on real-time traffic conditions. In this context, studies [32,37] incorporate congestion avoidance techniques to enhance their solutions. Specifically, ref. [32] integrates a congestion-avoidance mechanism that significantly improves the proposed algorithm's performance in terms of average latency. Similarly, ref. [37] designs its algorithm for evaluating current link congestion times, enabling intelligent link weighting and the selection of optimal paths.

Studies [39,42] combine Routing Optimization with network reliability. In [39], the authors propose a new routing algorithm for SDN-IoT environments to ensure reliable communication, consequently reducing latency in heterogeneous IoT applications. The approach in [42] introduces an algorithm for multi-hop SD-IoT networks that balances the trade-off between selecting paths with maximum reliability and those with minimal latency.

Several other Routing Optimization strategies, namely [31,35,39,46], rely on traffic monitoring to achieve low-latency communication. Notable among them, ref. [39] integrates link reliability with traffic monitoring, while [46] employs IoT network monitoring to design a new routing protocol based on a multi-hop clustering technique to minimize latency. Study [35] analyzes SDN-IoT traffic to identify paths with minimum delay and leverages a multi-controller architecture to improve scalability. Furthermore, ref. [40] addresses scalability challenges through centralized and hybrid SD-IoT routing mechanisms, increasing the number of IoT nodes and evaluating their impact on latency and network utilization.

Load balancing strategies facilitate the efficient distribution of network traffic to mitigate congestion, reduce latency, and improve scalability. By leveraging SDN's centralized control and real-time traffic analysis capabilities, intelligent load-balancing algorithms can dynamically allocate traffic across the network. In [34,44], the authors present multicontroller-based LB solutions. The approach in [34] classifies traffic into delay-sensitive and non-delay-sensitive categories to optimize routing decisions accordingly. In contrast, [44] introduces a strategy based on a switch migration mechanism to balance traffic load. Both studies assess their solutions concerning low-latency performance and scalability. Furthermore, [31] proposes a Quality of Experience (QoE) driven approach that integrates Load Balancing and Routing Optimization to identify paths with minimal delay. Meanwhile, [33] performs Load Balancing in fog servers using an adaptive policy to achieve low latency and reduced energy consumption.

Resource Allocation strategy, which leverages the centralized control of SDN, facilitates efficient network and computing resources management to accommodate the dynamic demands of SDN-IoT environments. Several selected studies have implemented this strategy alongside task or computation offloading, such as in [27,38]. The approach proposed in [43] focuses on cache management to enhance scalability within ubiquitous SD-IoT networks. Meanwhile, ref. [45] addresses Resource Allocation at the control plane level to ensure low-latency communication. Some studies classified primarily under Resource Allocation also incorporate other optimization techniques, including Load Balancing in [33] and Routing Optimization in [37].

Service Provisioning strategy includes solutions that, through service management and intelligent provisioning mechanisms, adapt to the demands of SD-IoT networks to ensure seamless service delivery with low latency. Two studies stand out in this context: ref. [25] implements an application-aware and delay-sensitive service provisioning mechanism in SD-IoT; Ref. [41] focuses on managing and provisioning Fog services, which can adapt to significant changes in system load while maintaining low latency.

Flow Management strategy optimizes the control and forwarding of data flows in network management and forwarding devices to enhance performance in SDN-IoT environments. Studies [29,49] focus on OpenFlow/SDN rule management. Ref. [29] proposes a dynamic and personalized bandwidth allocation approach, emphasizing low latency. Ref. [49] implements early installation of OpenFlow rules in switches to minimize latency in IoT communications. Ref. [30] introduces a solution based on Flow/Rules placement and presents a mechanism for managing flow tables in SDN-IoT networks, targeting low latency and improved accuracy in OpenFlow rule processing. Ref. [26] employs a collaborative flow-identification mechanism to secure SD-IoT networks, improving low latency and energy consumption performance. Ref. [28] applies Flow Management to IoT video applications, where analysis of IoT video flows enables the adjustment and determination of optimized rules to achieve low latency.

The studies described in this section indicate the main strategies and approaches based on ML to deal with low latency and scalability requirements in SD-IoT networks. Routing Optimization and Flow Management strategies are particularly prominent, with numerous solutions and variations emerging in the literature. Furthermore, it is important to note that most of these strategies prioritize low latency, suggesting a gap in the existing literature. Specifically, there is a need for solutions that not only focus on low latency but also evaluate scalability requirements, which is an essential aspect of this SLR.

6.2. Exploring Experimentation Scenarios/Environments (SRQ-2)

This subsection addresses research question SRQ-2 by examining the experimental scenarios and environments employed to validate the proposals presented in the selected

studies within the MRQ. Table 10 provides a summary of these studies, detailing (i) the types of scenarios used; (ii) the total number of studies per scenario; (iii) the individual identification of studies associated with each scenario; and (iv) the suitability of the experimental scenario implemented with IoT concepts. The distribution of experimental environments across the selected studies is as follows: Surveillance systems (three studies), Smart City/Building (two studies), Generic Scenario (seventeen studies), Unmanned Aerial Vehicles (UAVs) (one study), Underwater IoT (one study), Mobile IoT (one study), and Industrial IoT (one study). Studies classified under the “Generic Scenario” category include experimentation environments that incorporate one or more IoT components—such as sensors, IoT gateways, or Fog nodes—but lack detailed specifications regarding the data transmitted or generated, as well as the influence of the operational environment on the proposed solution. This lack of specificity denotes a generalized IoT context for these scenarios.

Studies [28,31] conducted experiments in the context of surveillance systems. Specifically, ref. [28] explores an IoT video streaming application involving IoT devices and Edge servers, while [31] employs a similar setup using IoT video cameras and Edge Computing infrastructure. Other studies, such as [47,49], rely on datasets derived from IoT environments in smart city and smart building contexts. In [47], the authors utilize real-world vehicular traffic data to develop a Routing Optimization solution. Ref. [49] leverages sensor data from a smart building to manage IoT/OpenFlow data flow. Additional experimentation scenarios also appear in the literature. For instance, ref. [27] investigates an application involving UAV, IoT nodes, and SDN infrastructure. Study [44] focuses on underwater IoT systems, incorporating underwater wireless networks and multi-controller SDN architectures. In the domain of mobile IoT, ref. [33] presents an application that uses Android and iPhone devices and fog nodes deployed via containers to support Load Balancing and Resource Allocation. Ref. [25] evaluates a service provisioning solution using datasets related to Industrial IoT and smart city environments.

Additionally, 17 studies were classified under generic IoT scenarios. These studies were grouped into three categories based on their suitability for IoT environments, as presented in Table 1: (i) fully suitable for IoT deployment, (ii) generic scenarios with high potential for IoT adaptation, and (iii) generic scenarios with moderate potential for IoT adaptation. The “IoT Suitability” column in Table 10 reflects the degree of effort and adaptation required to implement each solution within an IoT context. This includes integrating specific technologies, communication protocols, physical components, datasets, and the overall system architecture. As shown in Table 1, among the 17 studies classified as generic IoT scenarios, 11 (64.7%) were deemed fully suitable for IoT environments. Three studies demonstrated high adaptation potential, while the remaining three exhibited moderate suitability for integration into IoT systems.

In the context of SRQ-2, this section examines the scenarios employed in the studies selected for the SLR. As shown in Table 10, most studies evaluate and validate their solutions using generic IoT-based experimental setups. This observation highlights a notable gap and a corresponding need for more specialized IoT scenarios that reflect diverse and realistic operating environments. Furthermore, the data suggest that emerging domains such as Industrial IoT, Smart Cities, and Mobile IoT remain underexplored within the MRQ scope, presenting opportunities for future investigations in these contexts.

Table 10. Categorization of the selected studies based on the experimental scenarios used for validation, including application domains and levels of IoT suitability. The Table highlights whether each scenario represents a domain-specific IoT use case or a generic environment with varying potential for IoT adaptation.

Scenario	Total	Studies	IoT Suitability
Generic	17	[26,35,36,38–40,42,43,45,46,48]	Total
		[29,37,41]	Generic scenarios with high potential for IoT adaptation
		[30,32,34]	Generic scenarios with moderate potential for IoT adaptation
Surveillance systems	2	[28,31]	Total
UAVs	1	[27]	Total
Underwater IoT	1	[44]	Total
Mobile IoT	1	[33]	Total
Smart City/Building	3	[25,47,49]	Total
Industrial IoT	1	[25]	Total

6.3. Exploring Component Fidelity and Complexity for Deploying Solutions in Real Environments (SRQ-3)

This subsection addresses the research question SRQ-3 by exploring, based on the studies selected in this SLR, the fidelity and complexity of deployment of the selected solutions in real-world environments. Specifically, we analyze the types of environments used for experimentation and explore the level of effort or adaptation required to replicate and deploy these solutions in practical, real-world settings.

Table 11 categorizes the studies included in this SLR based on the deployment environment used for experimentation. The data presented in Table 11 and Figure 6 show that most studies (23 studies, 92%) conducted experiments primarily through simulation. Analysis of these studies reveals that Mininet, NS-3/NS-2, and the Markov Decision Process (MDP) are the principal tools and techniques employed to construct experimental environments. Ref. [50], adopted in 13 studies, is an open-source network emulator that enables the creation of virtual nodes, switches, SDN/OpenFlow controllers, and links. It also supports executing applications and libraries designed for Linux and provides access to the underlying kernel and networking stack. NS-3, used in four studies, is a discrete-event simulator widely used in research involving traditional and SDN-based network simulations. It supports various modular extensions, broadening the scope of possible experimental configurations. In six studies, MDP was used to model network behavior, aiming to optimize Resource Allocation and routing strategies in proposed solutions. Integrating MDP within ML and SDN contexts enhances its effectiveness by enabling decision-making based on network state transitions and probabilities. This dynamic approach supports more adaptive and efficient network management.

Table 11. Classification of the selected studies according to the experimental deployment setup or simulation environment adopted for evaluation, including network simulators, analytical models, and real-world testbeds.

Deployment	Studies
Mininet	[26,29,32,35,38,39,42–47,49]
NS-3/NS-2	[30,34,44,48]
Real-world environment	[36,41]
Markov Decision Process	[27–29,33,34,38,40]
Others	[25,31,37,47]

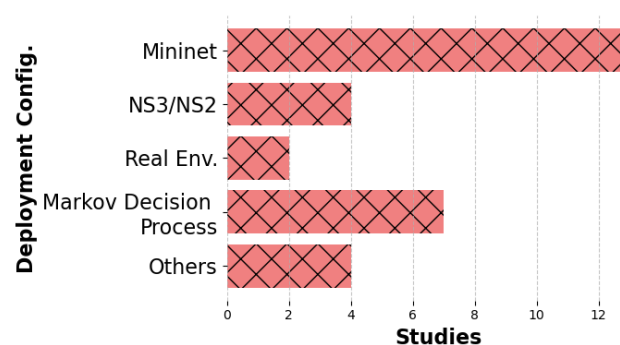


Figure 6. Distribution of the selected studies according to the type of experimental simulation environment used for validation, including network simulation (e.g., Mininet), simulation tools (e.g., NS-2/NS-3), analytical modeling (e.g., Markov Decision Processes).

In this subsection, we classify the selected studies based on the fidelity level of IoT component modeling within simulated experiments. Specifically, we group them into three categories: (i) high-fidelity component simulations, (ii) medium-fidelity component simulations, and (iii) low-fidelity component simulations. The following evaluation questions guide this categorization:

- **Hardware and Sensor Modeling:** Studies that describe, implement, or utilize realistic modeling of hardware components—such as energy consumption, data generation by IoT devices, and the execution of actual code or libraries on real or emulated devices—are classified as high-fidelity. In contrast, those that employ generic or abstract representations of sensors without accurate or detailed modeling are considered low-fidelity.
- **Communication Protocols in SD-IoT Networks:** Studies implementing or utilizing real communication protocols, such as Message Queuing Telemetry Transport (MQTT), OpenFlow, Constrained Application Protocol (CoAP), or Long Range Wide Area Network (LoRaWAN), are classified as high-fidelity. In contrast, studies that rely on abstract or generic representations of SDN-IoT communication without specifying concrete protocols are considered to exhibit low fidelity.
- **SD-IoT Network Modeling:** High-fidelity studies in SD-IoT networks typically involve using real SDN controllers, such as Ryu or POX, integrated with standard network protocol stacks, including TCP/IP and 5G. These experimental setups enable precise monitoring and retrieval of detailed performance metrics, such as latency, packet loss, and network congestion.

After analyzing the studies according to the categorization criteria discussed in the previous subsection, we have compiled Table 12. The data reveal that 17 (73.9%) of 23 studies with simulation-based experiments are classified as medium or low fidelity. In contrast,

six studies (26.1%) are categorized as high-fidelity modeling. Therefore, in the context of SRQ-3, and given that most studies are not classified as high-fidelity, the findings highlight the need for experiments that more accurately reflect the components of SD-IoT networks in Edge environments. It is important to emphasize that constructing experiments or models with high-fidelity components facilitates the adaptation and implementation of solutions in real-world environments, thus increasing the relevance and applicability of the proposed solutions.

Table 12. Studies categorized by the fidelity level of IoT component modeling within simulated experiments. Fidelity levels reflect the degree of realism in hardware modeling, communication protocols, and SD-IoT network components, indicating the potential for real-world deployment.

	Total	Studies
High-fidelity components	6 (26.1%)	[29,35,42,43,45,49]
Medium-fidelity components	9 (39.1%)	[26,30,32,33,38,39,44,46,47]
Low-fidelity components	8 (34.8%)	[25,27,28,31,34,37,40,48]

6.4. Exploring Machine Learning Methods and the Results Comparison (SRQ-4)

In this subsection, we address the secondary research question SRQ-4, which examines the ML methods employed in the studies selected for the context of the MRQ. We will discuss the types of ML methods utilized, their key characteristics, and how they are applied in the studies included in the SLR.

Based on the analysis of the selected studies, we identified and categorized the ML solutions into four distinct categories (see Figure 7): (i) Traditional Supervised Learning, which constitutes 24% of the studies. Algorithms in this category learn from labeled data, mapping inputs to outputs based on historical examples; (ii) Recurrent Neural Networks (RNNs), representing 12% of the studies. These algorithms process sequential data and maintain an internal memory to capture dependencies in time series and data streams; (iii) Convolutional Neural Networks (CNNs), accounting for 4% of the studies. These algorithms enable the analysis of spatial data, particularly images, by using convolutional layers to detect patterns and features; (iv) Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL), which cover 60% of the studies. These algorithms learn through trial and error by taking actions in a given scenario to maximize accumulated rewards.

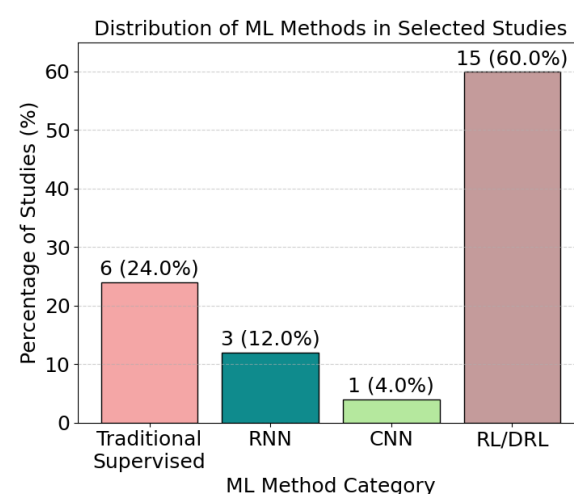


Figure 7. Classification of the selected studies according to the category of ML methods employed, including Traditional Supervised Learning, Recurrent Neural Networks, Convolutional Neural Networks, and Reinforcement/Deep Reinforcement Learning.

Figure 8 presents a heatmap that maps the association between ML methods and the strategies employed. Reinforcement Learning appears as the most prevalent approach across all strategic categories, particularly in Routing Optimization and Resource Allocation. In contrast, CNNs are less frequently adopted and tend to be applied to more narrowly defined problem scopes.

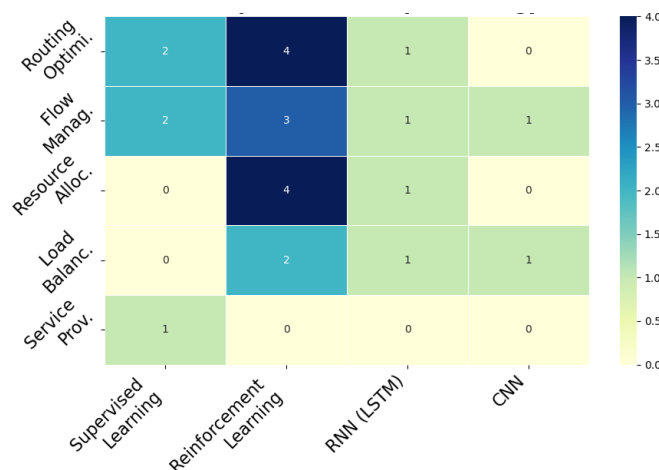


Figure 8. Heatmap illustrating the relationship between ML method categories and SD-IoT strategies. Columns represent ML methods (Traditional Supervised Learning, CNN, RNN/LSTM, and RL/DRL). At the same time, rows correspond to the main strategies: Routing Optimization, Flow Management, Resource Allocation, Load Balancing, and Service Provisioning. Darker cells indicate a higher concentration of studies applying a given ML method to a specific strategy.

Traditional Supervised Learning models are often lightweight and suitable for deployment at the Edge, enabling real-time decision-making with minimal delay when compared to cloud-based inference. Traditional Supervised Learning has been adopted across several studies in this SLR, particularly in the contexts of service provisioning [25], flow management [26,36], and Routing Optimization [42]. In [25], the authors apply and compare various Traditional Supervised Learning algorithms (see Table 13) using datasets from Industrial IoT and smart city environments. For each dataset, they identify the algorithm that yields the best performance. In [26], supervised learning techniques—including logistic regression, LightGBM, decision tree, and random forest—are employed to implement a flow identification mechanism for IoT traffic in SD-IoT networks utilizing the OpenFlow protocol. Similarly, ref. [36] applies a decision tree algorithm for flow identification within SD-IoT networks that leverages the Programming Protocol-Independent Packet Processors (P4) protocol. Meanwhile, ref. [42] implements the K-Nearest Neighbors (KNN) algorithm to detect potential link failures, enhancing routing decision-making in a global SD-IoT architecture focused on link reliability.

Recurrent Neural Networks have been employed in the studies reviewed in this SLR in conjunction with Routing Optimization and Load Balancing [31], Resource Allocation [43], and Flow Management [49]. In [31], an LSTM network is applied to predict optimal routes based on the analysis of IoT data and video streams within an IoT surveillance scenario. Similarly, ref. [43] utilizes an LSTM model combined with Federated Learning to predict user data and QoS scores, thereby supporting a cache management solution in an SD-IoT network. Study [49] leverages Long Short-Term Memory (LSTM) to forecast IoT data streams, specifically temperature samples, to assist a rule scheduler in SD-IoT networks. Additionally, ref. [34] applies a CNN, using the Non-Deep Lightweight Parallel Network algorithm, to classify delay-sensitive IoT flows, thus enabling Management Flow and Load Balancing strategies.

Table 13. Summary of ML algorithms employed in the selected studies. This Table reports the frequency of each algorithm and the corresponding studies.

Machine Learning Algorithm	Total	Studies
LSTM (RNN)	3	[31,43,49]
Non-Deep Lightweight Parallel Network (CNN)	1	[34]
Q-learning (RL/DRL)	7	[29,30,32,33,45,46]
Deep Q-learning (RL/DRL)	1	[40]
SARSA (RL/DRL)	1	[44]
A3C (RL/DRL)	2	[27,38]
Proposed approach (RL/DRL)	1	[35]
Naive bayes (Trad. supervised learning)	3	[25,37,47]
Random Forest / Decision Tree (Trad. supervised learning)	3	[25,26,36]
KNN (Trad. supervised learning)	2	[25,42]
Neural Network (Trad. supervised learning)	1	[25]
Logistic Regression (Trad. supervised learning)	1	[26]
Light Gradient Boosting (Trad. supervised learning)	1	[26]

Reinforcement Learning and Deep Reinforcement Learning have been applied across the selected studies to address Load Balancing (two studies), Routing Optimization (four studies), Resource Allocation (four studies), and Flow Management (three studies) strategies. In the context of Load Balancing, ref. [33] employs the Q-Learning algorithm to support Load Balancing and fog workload distribution in a mobile IoT scenario, whereas [44] utilizes the SARSA algorithm to perform multicontroller Load Balancing in an underwater SD-IoT network. For Routing Optimization, the studies adopt Q-Learning [32,46] and Deep Q-Learning [40] algorithms to enable intelligent and multipath routing in generic IoT scenarios. Additionally, ref. [35] proposes a custom DRL approach combined with Federated Learning for Routing Optimization. Regarding Resource Allocation, Asynchronous Advantage Actor–Critic (A3C) algorithms are used for task scheduling and offloading in IoT-UAV scenarios ([27], also incorporating Federated Learning) and in generic IoT environments [38]. Q-Learning is also employed to manage the SD-IoT network control plane [45]. In the area of Flow Management, Q-Learning facilitates admission control for incoming IoT video flows, determines optimal processing locations for video frames [28], and supports flow placement decisions based on prediction mechanisms [30].

Figure 9 categorizes the studies and presents data regarding the methods used to compare the ML solutions or algorithms in the experiments. Based on the analysis, the solutions were organized into four categories: (i) Traditional algorithms—studies that compared results using the traditional algorithms listed in Table 13; (ii) DRL/RL/CNN/RNN algorithms—studies that compared their results with reinforcement learning algorithms, Recurrent Neural Networks, or Convolutional Neural Networks; (iii) Baseline approach—studies that compared their solutions with and without the proposed ML algorithm, without performing compar-

isons against other algorithms; and (iv) No comparison—studies that presented results using the proposed algorithm but did not conduct any comparative analysis.

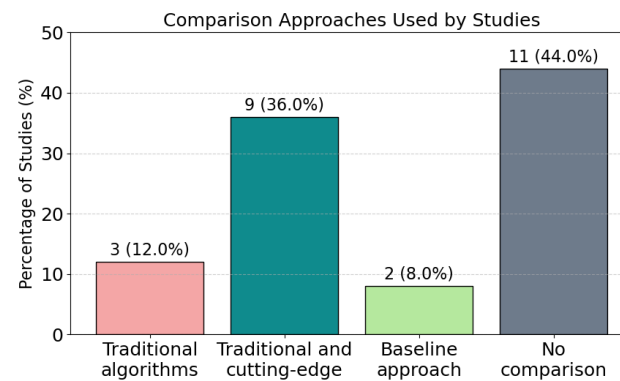


Figure 9. Categorization of the selected studies based on the type of comparative evaluation adopted.

Figure 9 shows that 44% of the studies in this SLR do not provide comparative analyses of the ML methods employed. In contrast, 36% of the studies compare their proposals with advanced algorithms such as DRL/RL/CNN/RNN, while 12% include comparisons with traditional ML techniques. Therefore, 48% of the studies perform comparative evaluations of their ML solutions against other algorithms. Additionally, 8% of the studies conduct baseline comparisons, assessing performance with and without the proposed solution.

Based on the data analyzed in this subsection, Reinforcement Learning and Deep Reinforcement Learning methods predominate among the selected studies, accounting for 60% of the approaches. This reflects a growing trend toward adopting trial-and-error learning techniques in SD-IoT environments, particularly for Routing Optimization, Load Balancing, and Resource Allocation. Although Traditional Supervised Learning methods are also present, their use appears more limited and typically focuses on specific tasks, including flow identification and Routing Optimization. Recurrent Neural Networks, especially LSTM architectures, also stand out in applications involving temporal prediction of IoT data.

A significant methodological gap emerged in the analysis of performance comparisons: 44% of the studies did not include any form of benchmarking against alternative approaches, undermining the objective assessment of the proposed solutions. Nevertheless, approximately half of the studies incorporated comparative evaluations, with 36% referencing advanced algorithms such as DRL, RL, CNNs, or RNNs. These findings underscore the need for more rigorous experimental methodologies in comparative analyses to strengthen the validation of proposed ML strategies within this systematic literature review.

7. Practical Applicability of ML Methods for Scalable SD-IoT Deployments

A key aspect of ensuring the real-world feasibility of ML-based SD-IoT solutions is understanding which ML approaches can effectively scale across latency-sensitive environments. Although the selected studies employ a diverse set of ML techniques, from Traditional Supervised Learning to Deep Reinforcement Learning, these methods exhibit considerable variation in computational cost, deployment complexity, and integration with SD-IoT infrastructures. By synthesizing the findings from SRQ-3 (fidelity and realism of deployments) and SRQ-4 (ML methods used), as presented in Table 12, Figures 7 and 8, it becomes possible to identify which methods currently show the most promise for scalable deployments in real SD-IoT environments.

In the selected studies, the most practically scalable techniques are Traditional Supervised Learning and Q-learning-based. As shown in Figure 7, supervised learning

models represent a substantial proportion of the selected studies, and their computational simplicity enables deployment directly on SDN controllers or Edge nodes without specialized hardware acceleration. Their suitability for real applications is further reinforced by the fact that supervised ML appears in several of the high-fidelity studies identified in Table 12, including [29,35,42,49]. Similarly, Q-learning-based solutions, also highlighted in the heatmap in Figure 8, demonstrate compatibility with real deployments due to their low memory requirements, fast convergence, and application in routing and flow optimization [32,40,46,48].

Table 14 summarizes the practical scalability profile of the ML methods examined in this SLR, integrating findings from Figures 7 and 8, and Table 12. In this context, the analysis indicates that scalable SD-IoT deployments prioritized lightweight ML models, particularly traditional supervised ML and Q-learning, while more complex DL architectures remain promising but require further investigation under high-fidelity, real-world conditions. These observations underscore the need for ongoing research on ML solutions for SD-IoT, complemented by high-fidelity experimental evaluations to bridge the gap between experimental results and deployable large-scale SD-IoT systems.

Table 14. Practical scalability of ML methods in SD-IoT deployments.

ML Method	Computation Requirements	Deployment Difficulty	Suitable Layer	Practical Scalability
Traditional Supervised ML	Low	Low	Edge/Fog/Controller	High
Q-Learning/Tabular RL	Low/Moderate	Low	Edge/Fog/Controller	High
Deep Reinforcement Learning	High	Moderate	Fog/Edge/Controller	Moderate
RNN/LSTM	High	Moderate/High	Fog	Low/Moderate
CNNs	High	High	Fog with GPU	Low

8. Common Architectural Trends and Divergent Aspects

This section summarizes the common architectural trends across the selected studies, highlighting how their design choices converge and diverge beyond the general use of SD-IoT and ML technologies. By examining the structural organization of the solutions, the mechanisms used to support decision-making, and the specific algorithmic strategies adopted, it becomes possible to identify recurring principles that shape current approaches. Thus, differences reveal the diversity of design goals, optimization criteria, and implementation methods explored in the literature.

8.1. Load Balancing Studies

Across the studies [31,33,34,44], a common point is the use of multi-stage decision pipelines, in which each solution is organized into several sequential modules rather than a single unified controller. These stages often include steps such as classification, prediction, prioritization, or migration analysis. For example, ref. [31] separates QoE classification from traffic forwarding, ref. [33] divides resource estimation and mobility management, ref. [34] employs user filtering, flow categorization, and hybrid balancing, ref. [44] separates switch migration from adaptive routing. This point indicates that multi-stage internal architectures are becoming an important design choice for load-balancing solutions in SD-IoT.

Another common feature is the differentiation of flows or tasks according to their performance requirements. Each system identifies categories, such as delay-sensitive flows, critical tasks, or high-priority traffic, and handles them differently to maintain low latency under high load. For example, ref. [34] classifies flows by sensitivity level using an ML algorithm (CNN-based), ref. [31] adjusts forwarding based on predicted QoE, ref. [33]

prioritizes delay-sensitive tasks during fog offloading, and [44] selects routes according to specific Quality of Service (QoS) demands. These mechanisms show a trend toward targeted, priority-based strategies rather than a uniform load distribution.

The studies also share the use of threshold-based triggers to activate adaptation procedures. Actions such as migration, rerouting, or resource reallocation in the Load Balancing context occur when key metrics surpass predefined limits. In [34], thresholds are used to redirect flows or instantiate virtual switches; in [44], they trigger migrations during periods of high controller load; in [31], traffic is redirected when QoE drops; and in [33], fog assignments are updated when resource or mobility constraints become critical.

8.2. Routing Optimization Studies

Across the selected Routing Optimization studies, several internal design similarities emerge that go beyond the broad use of ML and SD-IoT technologies. A first common aspect is the separation between offline preparation and online decision-making. Many works build training datasets or auxiliary prediction models offline and then use lightweight decision mechanisms during operation. Examples include offline route labeling in [31], reliability prediction in [39,42], and mobility-based destination prediction in [47]. This structure can reduce online computation while allowing complex intelligence to be precomputed.

Another recurring pattern is the use of multi-criteria performance objectives. In this context, instead of optimizing only one metric, the majority of studies combine several factors such as latency, reliability, load, and energy. Approaches like the multi-objective routing in [40,42], or the clustered energy-aware design of [46], demonstrate how QoS trade-offs are integrated into routing logic. Selected studies such as [33,37] also follow this trend by jointly optimizing execution time and energy consumption.

Despite these common aspects, the studies differ in how learning is integrated into the control of the solution. Some works use ML as a function approximator for an existing heuristic, such as the RNN in [31] or the Naive Bayes predictor in [47]. Others rely on direct policy learning, as seen in the Q-learning methods of [32,39,46]. DRL solutions present further variation: [35] uses federated learning across controllers, ref. [40] evaluates both centralized and distributed DRL modes, and [33,37] apply DRL directly to routing. In contrast, ref. [42] separates learning and optimization, using ML only to predict reliability and a multi-objective solver to select paths.

8.3. Resource Allocation Studies

A first similarity is the view of Resource Allocation as a dynamic and stochastic problem. Many studies model the environment using MDP or equivalent sequential decision frameworks, reflecting continuous changes in mobility, traffic, energy, and channel conditions. Works such as [33,38,48] explicitly build their algorithms on MDP formulations, while others, such as [31,37,44], still treat the environment as non-stationary and design policies that adapt over time.

As in Routing Optimization studies, the shared aspect is the use of multi-objective decision-making. Although latency is a core concern, it is generally not optimized alone. Instead, the studies evaluate and balance multiple goals, including energy efficiency ([27,33,38,48]), network reliability ([44]), QoS/QoE ([31,45]), or operational costs ([33]).

8.4. Flow Management Studies

The Flow Management studies converge on a common architectural idea: flows must be semantically understood before they are controlled. Different proposals implement this via explicit traffic or task classification modules. Rather than treating traffic uniformly, the solutions identify classes of flows—such as heterogeneous IoT traffic in [26], heavy or active flows in [30], delay-sensitive flows in [34], or high-priority jobs in [37]. Even security-

oriented ([36]) and video-oriented approaches ([28,31]) rely on flow-type identification before deciding how to manage them. This shows that semantic understanding of traffic is now an important mechanism in SD-IoT Flow Management solutions.

Another similarity is the adoption of hybrid proactive–reactive control strategies. In contrast to relying exclusively on reactive OpenFlow behavior, many works use predictions to proactively configure flow rules or allocate resources while still reacting to unexpected situations. Examples include mobility-aware rule placement in [30], multi-controller proactive and reactive balancing in [34], and proactive installation of OpenFlow rules in [49]. Works like [28,31] also evaluate routing and processing decisions before admitting new flows, forming hybrid decision loops.

9. Architectural Trends and Their Impact on Latency and Scalability

The selected studies reveal recurring architectural patterns that guide how ML-based SD-IoT solutions attempt to meet low latency and scalability requirements. Although each work uses different algorithms and deployment settings, their architectures share common ideas about distributing intelligence, managing flows, and adapting decisions in dynamic network conditions. In addition, the selected studies often do not explicitly define their architectural model. We identified common patterns by interpreting their system descriptions. These inferred tendencies provide a structured basis for analyzing how architectural decisions influence latency, scalability, and the ML methods used.

9.1. Flow Differentiation and Prioritization

Another common architectural trend is flow differentiation. Many solutions classify network flows using supervised ML models so that traffic with different requirements can be treated differently. For example, latency-sensitive flows may receive faster routing or dedicated resources, while background flows are handled with lower priority. Supervised ML models, such as Decision Tree and Random Forest in [26,36], or Naïve Bayes in [37], enable rapid classification of IoT flows into categories such as delay-sensitive, bandwidth-intensive, or background traffic.

This approach improves latency by reducing waiting time and ensuring quick processing of important flows. It also supports scalability because the controller does not need to treat all flows equally, reducing its workload. The main limitation is that misclassifications or heavy models may increase overhead at the Edge and lead to wrong routing decisions.

9.2. Multi-Stage Decision Pipelines

A prominent architectural trend is the adoption of multi-stage decision pipelines, where intelligence is distributed across layered components such as IoT devices, Fog/Edge nodes, and SDN controllers. These multi-stage pipelines help reduce latency by allowing faster, local decisions (often based on RL or DRL) without always contacting the SDN controller. Examples include RL-driven routing combined with centralized optimization ([32,35,40,46]), and Flow Management ([30]).

From a scalability perspective, the distribution of tasks across the network prevents the controller from becoming overloaded as the number of IoT devices increases. However, these architectures also face challenges: if the coordination between layers is not well handled, the system may produce inconsistent decisions or delayed reactions under heavy traffic.

9.3. Threshold-Based Activation

Several works use threshold-based triggers to decide when ML models or control mechanisms should run. Instead of constantly monitoring or recalculating routes, the

system activates decisions only when certain conditions are met (e.g., queue size, link quality, congestion level).

This approach helps reduce latency by ensuring that corrective actions occur only when needed, without introducing delays from unnecessary computation [37,40,48]. Threshold mechanisms also strengthen scalability because they limit the number of times ML modules need to be executed. The main challenge is that thresholds are often manually configured; if set incorrectly, they can trigger actions too late or too frequently.

9.4. Interplay Between Architectural Trends and ML Methods

Across the reviewed works, different ML techniques naturally align with different architectural choices:

- RL/DRL approaches in [29,30,32,44,46] strengthen multi-stage pipelines by enabling continuous adaptation to network dynamics.
- Supervised Learning models in [26,36,37,42] operationalize flow differentiation through lightweight inference suited for Edge deployment.
- The RNN/LSTM predictors in [46,49] can complement threshold-based triggers by forecasting traffic patterns or flow demand, allowing proactive rule placement and reducing reactive delays.

10. Identified Gaps and Challenges

Although several solutions based on ML techniques have been proposed to optimize SD-IoT networks in the Edge, particularly regarding low latency and scalability requirements, the analysis of the studies in this SLR identified important gaps that challenge the consolidation of these approaches in real and dynamic scenarios. Thus, based on the analysis of the studies in this SLR, in the following subsections, we discuss these gaps by categorizing the main obstacles/shortcomings identified in the analyzed studies and highlighting directions for future research.

10.1. Scalability Solutions

The analysis of the studies included in this systematic literature review reveals a predominant focus on addressing the low-latency requirements of SD-IoT networks, with comparatively less emphasis on scalability. Most of the selected works concentrate on Routing Optimization and Flow Management strategies, aiming to minimize response times and prioritize delay-sensitive traffic. In contrast, relatively few studies explicitly target scalability concerns. Although some studies—such as [34,35,40,43] acknowledge or incorporate scalability-related aspects (e.g., employing multi-controller architectures or accommodating increases in network nodes and traffic), such efforts remain limited. This imbalance highlights a significant research gap, particularly given the projected exponential growth in connected devices and the urgent need for solutions capable of sustaining performance under increasing load.

The lack of robust AI-based solutions for scalability in SD-IoT environments may limit the applicability of proposed approaches in real-world, dynamic scenarios such as smart cities, industrial systems, and critical infrastructure, where networks must adapt efficiently to abrupt traffic fluctuations. Based on the analysis conducted in this SLR, we observe a clear need for a shift in research focus toward strategies that concurrently address low-latency and scalability requirements in a balanced and integrated manner.

10.2. SD-IoT Deployment and Scenarios

The predominance of generic scenarios, identified in 17 studies analyzed, reflects a trend toward constructing simplified environments that may not fully capture the com-

plexity and heterogeneity inherent in IoT ecosystems. These scenarios often lack detailed specifications regarding device types, traffic patterns, and specific operating conditions, which may hinder the ability of experiments to reflect the real-world challenges encountered in SD-IoT settings accurately. Consequently, solutions validated in these generic environments may exhibit reduced or unpredictable performance when deployed in real-world contexts.

The need for more context-specific experimentation is evident, given the broad range of domains where IoT can be applied—such as healthcare, agriculture, logistics, and smart environments. Furthermore, environments with particular requirements, such as surveillance systems ([28,31]) and smart cities ([25,27]), demand solutions tailored to their unique characteristics. The lack of such scenarios in most of the studies analyzed highlights a recurring challenge in this systematic literature review: the difficulty in accessing more representative experimental environments and the limited resources for simulating complex conditions.

To enable better cross-study comparisons and improve reproducibility, future works should consider adopting publicly available and standardized IoT datasets in their experimental evaluations. Examples include the IoT-23 dataset [51], the UCI ML Repository (IoT subsets) [52], and the CityPulse project [53]. These datasets provide realistic traces from smart city and urban sensing environments and can help assess the effectiveness of ML-based solutions under diverse conditions.

10.3. Federated Learning

Federated Learning (FL) offers a promising approach for integrating ML techniques into SD-IoT networks in a distributed manner, preserving data privacy and reducing communication overhead with central servers. However, despite its potential, the adoption of FL remains in its early stages within the studies reviewed in this RSL. It was identified in only three works ([27,35,43]), all of which use FL as a complementary technique alongside other algorithms such as A3C, DRL, and LSTM. This limitation highlights a significant gap in exploring FL as a central solution to scalability and low-latency challenges.

Although only three studies among the selected works adopted Federated Learning, this approach presents significant potential for addressing scalability and privacy in SD-IoT environments. FL enables decentralized model training by keeping raw data on Edge devices, thus preserving user privacy and reducing communication overhead with central servers. However, FL also introduces challenges related to communication cost and data heterogeneity. The increased number of local model updates and the need for frequent synchronization can result in substantial communication overhead, particularly in networks with limited bandwidth or unreliable connections [54].

Emerging frameworks such as FedML [55] and TensorFlow Federated [56] are being developed to support federated learning in resource-constrained environments. These tools provide libraries and simulation platforms tailored to IoT and Edge applications, representing a promising direction for future research on distributed intelligence in SD-IoT infrastructures.

10.4. Security-Aware ML Solutions for SD-IoT

Cybersecurity is an important requirement in SD-IoT systems, since attacks on devices, controllers, or communication flows can directly increase latency and reduce the system's ability to scale. Recent studies in IoT and smart-home security show that high-quality and structured cyber-threat intelligence is essential for ML-based detection mechanisms to operate effectively. For example, recent work on IoT cybersecurity, but not applicable and evaluated in SD-IoT infrastructures, demonstrates that AI-based techniques, such as RNN models combined with optimized feature selection, can significantly accelerate the

detection and classification of cyberattacks, enabling faster and more adaptive protection mechanisms in distributed IoT environments [57,58].

Despite the importance of cybersecurity for maintaining reliable and responsive SD-IoT infrastructures, and the study [26], which only superficially addresses security through Flow Management, our review did not find any studies that integrate Security-Aware ML techniques with an explicit focus on low latency or scalability. This reveals an additional research gap: current work does not yet address how ML-based security can be incorporated into SD-IoT with a focus on improving latency and scalability. Future research should, therefore, explore lightweight, Edge-friendly ML security mechanisms and threat-intelligence integration strategies that protect SD-IoT systems while preserving low latency and scalable operation.

11. Limitations of This Study

In this section, we discuss possible threats to the validity of this SLR, according to the following categories: construct validity, internal validity, external validity, and reliability [59].

Construct validity addresses issues related to study design, specifically examining the extent to which the research accurately represents the concepts intended by the researchers and aligns with the research questions. This study employed automated and manual search strategies across multiple databases to mitigate threats to construct validity. This approach aimed to broaden the search scope and more accurately capture the breadth of the topics outlined by the SRQs. Additionally, the application of predefined inclusion and exclusion criteria ensured that only studies aligned with the review's objectives, namely, those offering solutions relevant to the MRQ, were included. The involvement of multiple researchers in the screening and discussion phases also promoted consistency in conceptual representation and helped minimize individual interpretative biases.

Regarding internal validity, which pertains to the consistency between the extracted data and the interpretations made, we acknowledge the potential risk of individual bias in analyzing the studies. We implemented several review steps to minimize errors during the selection and extraction of information, including multiple readings of the selected papers. This approach aimed to ensure the inclusion of the most comprehensive studies, thereby enhancing the reliability of our conclusions. A review protocol was also established and adhered to throughout the study to maintain methodological rigor.

External validity addresses the extent to which the results of the selected studies can be generalized to other application scenarios and domains. In this SLR, we focus on studies that involve practical applications and provide detailed descriptions of experiments, thereby enhancing the applicability of the findings within the investigated domain. However, we note that many of the selected studies are limited to simulated scenarios with low fidelity of components, which may restrict the generalizability of the results.

Reliability refers to the degree to which this study can be replicated, enabling other researchers to achieve the same results. To address this concern, we explicitly defined the research procedures to enhance replicability. However, the analysis and categorization of the data partially rely on the reviewers' interpretations, making the results susceptible to subjective variations.

12. Conclusions

In this study, we presented the results of a systematic mapping review of the literature, aiming to evaluate the current state-of-the-art research on SD-IoT supported by ML solutions, with a particular emphasis on low latency and scalability requirements. The review was conducted in accordance with established best practices found in the literature,

resulting in the systematic selection of 25 studies from an initial pool of 1719 publications (Steps 1 and 2 in Figure 2), retrieved primarily from five academic databases: ACM, IEEE Xplore, ScienceDirect, and Springer.

Based on our research and subsequent data analysis, we categorized the studies according to the following criteria: main strategies (Section 6.1), IoT scenarios and the suitability of the proposals for real environments (Section 6.2), fidelity of the experiments and complexity of solution implementation (Section 6.3), and characterization of the ML methods employed in the studies included in this SLR (Section 6.1). These sections provided the foundation for addressing the MRQ and the four SRQs.

Finally, this SLR has provided essential insights into the strategies, deployments, scenarios, and ML methods employed to achieve low latency and scalability in SD-IoT architectures. At the same time, the review revealed several important gaps that should guide future research efforts. First, we identified an imbalance in the literature: although many proposals focus on reducing latency, significantly fewer works rigorously evaluate scalability, especially under realistic multi-controller or large-scale IoT conditions. Second, most studies still rely on simulation-based experiments, highlighting the need for more realistic and reproducible evaluations that better reflect real-world wireless variability, device heterogeneity, and Edge resource constraints. Third, two promising directions, Federated Learning and Security-Aware ML—remain notably underexplored.

These observations emphasize that, beyond consolidating the current state of the art, this SLR also provides practical guidance for future research, identifying where new methods, more robust evaluations, and emerging ML paradigms can meaningfully advance the field.

Author Contributions: E.P.B.J.: Conceptualization, methodology, writing of the original draft, review. A.S.: writing of the original draft, review. M.P.: writing of the original draft, review. G.F.: Conceptualization, methodology, writing of the original draft, review, supervision. C.P.: Conceptualization, methodology, writing of the original draft, review, supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This study was partly supported by FAPESB INCITE PIE0002/2022 grant. Additionally, the authors acknowledge the financial support provided by the Pró-Reitoria de Pesquisa e Pós-Graduação (PRPPG) of the Federal University of Bahia through the “Programa de Apoio à Publicações Científicas, Edital 11/2026”, which covered the article processing charges for this publication. The authors also thank the CNPq, CAPES, and FAPESB organizations for their support of the Graduate Program in Computer Science at the Federal University of Bahia.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

Acronym	Description
AIoT	Artificial Intelligence of Things
IoT	Internet of Things
SDN	Software-Defined Networking
SD-IoT	Software-Defined Internet of Things
NFV	Network Function Virtualization
QoS	Quality of Service
QoE	Quality of Experience
UAV	Unmanned Aerial Vehicle

IoUT	Internet of Underwater Things
MQTT	Message Queuing Telemetry Transport
CoAP	Constrained Application Protocol
LoRaWAN	Long Range Wide Area Network
MDP	Markov Decision Process
P4	Programming Protocol-Independent Packet Processors
ML	Machine Learning
DRL	Deep Reinforcement Learning
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
KNN	k-Nearest Neighbors
FL	Federated Learning
A3C	Asynchronous Advantage Actor-Critic
SLR	Systematic Literature Review
PICOC	Population, Intervention, Comparison, Outcome, Context
MRQ	Main Research Question
SRQ	Secondary Research Question

References

1. Saxena, M.; Srivastava, S.; Dwivedi, V.K.; Chitranshi, R.; Mishra, P.K. Minimizing End-to-End Latency in Edge Computing-Enabled IoT Networks Through Edge-to-Edge Resource Allocation. In *Proceedings of the 2024 International Conference on IoT, Communication and Automation Technology (ICICAT), Gorakhpur, India, 23–24 November 2024*; IEEE: New York, NY, USA, 2024; pp. 1512–1517. [\[CrossRef\]](#)
2. Aboubakar, M.; Kellil, M.; Roux, P. A review of IoT network management: Current status and perspectives. *J. King Saud Univ.—Comput. Inf. Sci.* **2021**, *34*, 4163–4176. [\[CrossRef\]](#)
3. Chang, Z.; Liu, S.; Xiong, X.; Cai, Z.; Tu, G. A Survey of Recent Advances in Edge-Computing-Powered Artificial Intelligence of Things. *IEEE Internet Things J.* **2021**, *8*, 13849–13875. [\[CrossRef\]](#)
4. Ghanbari, H.; Khayyambashi, M.R.; Movahedinia, N. Improving Fog Computing Scalability in Software Defined Network Using Critical Requests Prediction in IoT. In *Proceedings of the 2021 12th International Conference on Information and Knowledge Technology (IKT), Babol, Iran, 14–16 December 2021*; IEEE: New York, NY, USA, 2021; pp. 6–10. [\[CrossRef\]](#)
5. Batista, E.; Figueiredo, G.; Prazeres, C. Load balancing between fog and cloud in fog of things based platforms through software-defined networking. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 7111–7125. [\[CrossRef\]](#)
6. Anantharaman, P.; Palani, K.; Smith, S. Scalable Identity and Key Management for Publish-Subscribe Protocols in the Internet-of-Things. In *Proceedings of the 9th International Conference on the Internet of Things, Bilbao, Spain, 22–25 October 2019*; IoT '19; Association for Computing Machinery: New York, NY, USA, 2019. [\[CrossRef\]](#)
7. Arellanes, D.; Lau, K.K. Analysis and Classification of Service Interactions for the Scalability of the Internet of Things. In *Proceedings of the 2018 IEEE International Congress on Internet of Things (ICIOT), San Francisco, CA, USA, 2–7 July 2018*; IEEE: New York, NY, USA, 2018; pp. 80–87. [\[CrossRef\]](#)
8. Rafique, W.; Qi, L.; Yaqoob, I.; Imran, M.; Rasool, R.U.; Dou, W. Complementing IoT Services Through Software Defined Networking and Edge Computing: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1761–1804. [\[CrossRef\]](#)
9. Bhayo, J.; Hameed, S.; Shah, S.A. An Efficient Counter-Based DDoS Attack Detection Framework Leveraging Software Defined IoT (SD-IoT). *IEEE Access* **2020**, *8*, 221612–221631. [\[CrossRef\]](#)
10. Siddiqui, S.; Hameed, S.; Shah, S.A.; Ahmad, I.; Aneiba, A.; Draheim, D.; Dustdar, S. Toward Software-Defined Networking-Based IoT Frameworks: A Systematic Literature Review, Taxonomy, Open Challenges and Prospects. *IEEE Access* **2022**, *10*, 70850–70901. [\[CrossRef\]](#)
11. Kitchenham, B.A.; Charters, S. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*; Technical Report EBSE 2007-001; Keele University and Durham University Joint Report: Staffordshire, UK, 2007.
12. do Carmo Machado, I.; McGregor, J.D.; Cavalcanti, Y.C.; de Almeida, E.S. On strategies for testing software product lines: A systematic literature review. *Inf. Softw. Technol.* **2014**, *56*, 1183–1199. [\[CrossRef\]](#)
13. Petticrew, M.; Roberts, H. *Systematic Reviews in the Social Sciences: A Practical Guide*; Wiley: Hoboken, NJ, USA, 2006; Volume 11. [\[CrossRef\]](#)
14. Zormati, M.A.; Lakhlef, H.; Ouni, S. Review and analysis of recent advances in intelligent network softwarization for the Internet of Things. *Comput. Netw.* **2024**, *241*, 110215. [\[CrossRef\]](#)

15. Amin, R.; Rojas, E.; Aqdus, A.; Ramzan, S.; Casillas-Perez, D.; Arco, J.M. A Survey on Machine Learning Techniques for Routing Optimization in SDN. *IEEE Access* **2021**, *9*, 104582–104611. [\[CrossRef\]](#)
16. Singh, R.; Gill, S.S. Edge AI: A survey. *Internet Things Cyber-Phys. Syst.* **2023**, *3*, 71–92. [\[CrossRef\]](#)
17. Jurado-Lasso, F.F.; Marchegiani, L.; Jurado, J.F.; Abu-Mahfouz, A.M.; Fafoutis, X. A Survey on Machine Learning Software-Defined Wireless Sensor Networks (ML-SDWSNs): Current Status and Major Challenges. *IEEE Access* **2022**, *10*, 23560–23592. [\[CrossRef\]](#)
18. Alonso, R.S.; Prieto, J.; La Prieta, F.d.; Rodríguez-González, S.; Corchado, J.M. A Review on Deep Reinforcement Learning for the management of SDN and NFV in Edge-IoT. In *Proceedings of the 2021 IEEE Globecom Workshops (GC Wkshps), Madrid, Spain, 7–11 December 2021*; IEEE: New York, NY, USA, 2021; pp. 1–6. [\[CrossRef\]](#)
19. Rezaee, M.R.; Abdul Hamid, N.A.W.; Hussin, M.; Zukarnain, Z.A. Fog Offloading and Task Management in IoT-Fog-Cloud Environment: Review of Algorithms, Networks, and SDN Application. *IEEE Access* **2024**, *12*, 39058–39080. [\[CrossRef\]](#)
20. Lone, K.; Sofi, S.A. A review on offloading in fog-based Internet of Things: Architecture, machine learning approaches, and open issues. *High-Confid. Comput.* **2023**, *3*, 100124. [\[CrossRef\]](#)
21. Chen, L.; Babar, M.A.; Zhang, H. Towards an Evidence-Based Understanding of Electronic Data Sources. In *Proceedings of the International Conference on Evaluation & Assessment in Software Engineering, Staffordshire, UK, 12–13 April 2010*.
22. Kitchenham, B.A.; Brereton, P.; Turner, M.; Niazi, M.K.; Linkman, S.; Pretorius, R.; Budgen, D. Refining the systematic literature review process—Two participant-observer case studies. *Empir. Softw. Eng.* **2010**, *15*, 618–653. [\[CrossRef\]](#)
23. Jalali, S.; Wohlin, C. Systematic literature studies: Database searches vs. backward snowballing. In *Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, Lund, Sweden, 19–20 September 2012*; IEEE: New York, NY, USA, 2012; pp. 29–38. [\[CrossRef\]](#)
24. Dybå, T.; Dingsøyr, T. Empirical studies of agile software development: A systematic review. *Inf. Softw. Technol.* **2008**, *50*, 833–859. [\[CrossRef\]](#)
25. Saha, R.; Ahmed, N.; Misra, S. Dynamic Fog Intelligence with Flow Control for Green Internet of Things. In *Proceedings of the GLOBECOM 2022—2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4–8 December 2022*; IEEE: New York, NY, USA, 2022; pp. 1–6. [\[CrossRef\]](#)
26. Ahmed, N.; Misra, S. Collaborative Flow-Identification Mechanism for Software-Defined Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 3457–3464. [\[CrossRef\]](#)
27. Qureshi, K.I.; Wang, L.; Xiong, X.; Lodhi, M.A. Asynchronous Federated Learning for Resource Allocation in Software-Defined Internet of UAVs. *IEEE Internet Things J.* **2024**, *11*, 20899–20911. [\[CrossRef\]](#)
28. Shahrbabaki, P.P.; Coutinho, R.W.L.; Shayan, Y.R. SDNQ: A Novel SDN Controller for the Management of IoT Video Flows in the Edge/Cloud Continuum. In *Proceedings of the ICC 2024—IEEE International Conference on Communications, Denver, CO, USA, 9–13 June 2024*; IEEE: New York, NY, USA, 2024; pp. 5419–5424. [\[CrossRef\]](#)
29. Qadeer, A.; Lee, M.J.; Tsukamoto, K. Flow-Level Dynamic Bandwidth Allocation in SDN-Enabled Edge Cloud using Heuristic Reinforcement Learning. In *Proceedings of the 2021 8th International Conference on Future Internet of Things and Cloud (FiCloud), Virtual, 23–25 August 2021*; IEEE: New York, NY, USA, 2021; pp. 1–10. [\[CrossRef\]](#)
30. Huang, G.; Ullah, I.; Huang, H.; Kim, K.T. Predictive mobility and cost-aware flow placement in SDN-based IoT networks: A Q-learning approach. *J. Cloud Comput.* **2024**, *13*, 26. [\[CrossRef\]](#)
31. Nazemi Absardi, Z.; Javidan, R. A QoE-driven SDN traffic management for IoT-enabled surveillance systems using deep learning based on edge cloud computing. *J. Supercomput.* **2023**, *79*, 19168–19193. [\[CrossRef\]](#)
32. Hassen, H.; Meherzi, S.; Jemaa, Z. Improved Exploration Strategy for Q-Learning Based Multipath Routing in SDN Networks. *J. Netw. Syst. Manag.* **2024**, *32*, 25. [\[CrossRef\]](#)
33. Lakhan, A.; Mohammed, M.A.; Obaid, O.I.; Chakraborty, C.; Abdulkareem, K.H.; Kadry, S. Efficient deep-reinforcement learning aware resource allocation in SDN-enabled fog paradigm. *Autom. Softw. Eng.* **2022**, *29*, 20. [\[CrossRef\]](#)
34. Tiwana, P.S.; Singh, J. Triple Tier Framework for Intellectual Edge Assisted Multicontroller Load Balancing in SDN. *SN Comput. Sci.* **2024**, *5*, 960. [\[CrossRef\]](#)
35. Zormati, M.A.; Lakhlef, H.; Ouni, S. Routing Optimization Based on Distributed Intelligent Network Softwarization for the Internet of Things. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, Avila, Spain, 8–12 April 2024*; SAC '24; Association for Computing Machinery: New York, NY, USA, 2024; pp. 1757–1764. [\[CrossRef\]](#)
36. Nguyen, H.N.; Nguyen, M.D.; Montes de Oca, E. A Framework for In-network Inference using P4. In *Proceedings of the 19th International Conference on Availability, Reliability and Security, Vienna, Austria, 30 July 2024–2 August 2024*; ARES '24; Association for Computing Machinery: New York, NY, USA, 2024. [\[CrossRef\]](#)
37. Sarma, B.; Kumar, R.; Tuithung, T. Machine learning enabled network and task management in SDN based Fog architecture. *Comput. Electr. Eng.* **2023**, *108*, 108705. [\[CrossRef\]](#)
38. Sellami, B.; Hakiri, A.; Ben Yahia, S. Deep Reinforcement Learning for energy-aware task offloading in join SDN-Blockchain 5G massive IoT edge network. *Future Gener. Comput. Syst.* **2022**, *137*, 363–379. [\[CrossRef\]](#)

39. Jisi, C.; hee Roh, B.; Ali, J. Reliable paths prediction with intelligent data plane monitoring enabled reinforcement learning in SD-IoT. *J. King Saud Univ.—Comput. Inf. Sci.* **2024**, *36*, 102006. [\[CrossRef\]](#)
40. Cong, P.; Zhang, Y.; Liu, Z.; Baker, T.; Tawfik, H.; Wang, W.; Xu, K.; Li, R.; Li, F. A deep reinforcement learning-based multi-optimality routing scheme for dynamic IoT networks. *Comput. Netw.* **2021**, *192*, 108057. [\[CrossRef\]](#)
41. Fröhlich, P.; Gelenbe, E.; Nowak, M.P. Smart SDN Management of Fog Services. In *Proceedings of the 2020 Global Internet of Things Summit (GloTS), Dublin, Ireland, 3–5 June 2020*; IEEE: New York, NY, USA, 2020; pp. 1–6. [\[CrossRef\]](#)
42. Akbar, A.; Ibrar, M.; Jan, M.A.; Bashir, A.K.; Wang, L. SDN-Enabled Adaptive and Reliable Communication in IoT-Fog Environment Using Machine Learning and Multiobjective Optimization. *IEEE Internet Things J.* **2021**, *8*, 3057–3065. [\[CrossRef\]](#)
43. Balasubramanian, V.; Aloqaily, M.; Reisslein, M.; Scaglione, A. Intelligent Resource Management at the Edge for Ubiquitous IoT: An SDN-Based Federated Learning Approach. *IEEE Netw.* **2021**, *35*, 114–121. [\[CrossRef\]](#)
44. Shi, Y.; Yang, Q.; Huang, X.; Li, D.; Huang, X. An SDN-Enabled Framework for a Load-Balanced and QoS-Aware Internet of Underwater Things. *IEEE Internet Things J.* **2023**, *10*, 7824–7834. [\[CrossRef\]](#)
45. Moslehi, M.; Ebrahimpour, H.; Goli, S.; Taji, R. A QoS Optimization Technique with Deep Reinforcement Learning in SDN-Based IoT. *Majlesi J. Electr. Eng.* **2021**, *15*, 105–113. [\[CrossRef\]](#)
46. Ouhab, A.; Abreu, T.; Slimani, H.; Mellouk, A. Energy-efficient clustering and routing algorithm for large-scale SDN-based IoT monitoring. In *Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020*; IEEE: New York, NY, USA, 2020; pp. 1–6. [\[CrossRef\]](#)
47. El-garoui, L.; Pierre, S.; Chamberland, S. A New SDN-Based Routing Protocol for Improving Delay in Smart City Environments. *Smart Cities* **2020**, *3*, 1004–1021. [\[CrossRef\]](#)
48. Banerjee, A.; Sufian, A. Reinforcement Learning Based Transmission Range Control (RL-TRC) in SD-WSN with Moving Sensors. *arXiv* **2020**, arXiv:2005.08215.
49. Batista, E.; Mello, B.; Silva, E.; Canário, J.P.; Rios, R.; Dustdar, S.; Figueiredo, G.; Prazeres, C. A New Intelligent Scheduler to Improve Reactive OpenFlow Communication in SDN-Based IoT Data Streams. *Discov. Internet Things* **2024**, *4*, 15. [\[CrossRef\]](#)
50. Mininet. Mininet Network Emulator. 2024. Available online: <http://mininet.org/overview/> (accessed on 21 January 2024).
51. Garcia, S.; Parmisano, A.; Erquiaga, M.J. *IoT-23: A Labeled Dataset with Malicious and Benign IoT Network Traffic*; Zenodo: Geneva, Switzerland, 2020. [\[CrossRef\]](#)
52. UCI. UCI Machine Learning Repository. 2025. Available online: <https://archive.ics.uci.edu/> (accessed on 10 May 2025).
53. CityPulse. CityPulse Project. 2025. Available online: <https://github.com/tianyuanhuang/citypulse> (accessed on 11 May 2025).
54. Ghimire, B.; Rawat, D.B. Recent Advances on Federated Learning for Cybersecurity and Cybersecurity for Federated Learning for Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 8229–8249. [\[CrossRef\]](#)
55. FedML. FedML—Federated Learning Easily, Scalably and Economically. 2025. Available online: <https://fedml.ai> (accessed on 12 May 2025).
56. TensorFlow. TensorFlow Federated. 2025. Available online: <https://www.tensorflow.org/federated> (accessed on 11 May 2025).
57. Rashed, M.; Viso, I.T.D.; González-Tablas, A.I. A Comparison of Cyber Intelligence Platforms in the Context of IoT Devices and Smart Homes. *Electronics* **2025**, *14*, 4503. [\[CrossRef\]](#)
58. Allafi, R.; Alzahrani, I.R. Enhancing Cybersecurity in the Internet of Things Environment Using Artificial Orca Algorithm and Ensemble Learning Model. *IEEE Access* **2024**, *12*, 63282–63291. [\[CrossRef\]](#)
59. Wohlin, C.; Runeson, P.; Höst, M.; Ohlsson, M.; Regnell, B.; Wesslén, A. Analysis and Interpretation. In *Experimentation in Software Engineering*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 153–184. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.