Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Data-Driven Unified Taxonomy and Benchmarking Framework for Machine Learning-Based Task Scheduling in Edge–Cloud Computing

Nagwa Elmobark

eng_nagwaelmobark@yahoo.com

Mansoura University

Adel Mounir Said

National Telecommunication Institute (NTI)

Research Article

Keywords: Edge–Cloud Computing, Task Scheduling, Machine Learning, Benchmarking, Unified Taxonomy, Evaluation Framework

Posted Date: June 5th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9924991/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

A Data-Driven Unified Taxonomy and Benchmarking Framework for Machine Learning-Based Task Scheduling in Edge–Cloud Computing

Nagwa Elmobark[1,2]*, Adel Mounir Said[3],

[1*] Researcher, Department of Computer Science, University of Mansoura, Mansoura, Egypt, eng_nagwaelmobark@yahoo.com, ORCID: 0000-0002-6757-088X

[2*]lecturer, Smart University of Science and Technology, UAE, Nagwa.e@sust.ae

[3*]Associate Professor, National Telecommunication Institute (NTI), Switching Department, Cairo, Egypt, adel.mounir@nti.sci.eg, ORCID: 0000-0002-3388-7918

Abstract

Machine learning (ML) scheduling of tasks has become a fundamental research focus in edge-cloud computing, and seeks to optimize latency, energy, and resource usage in distributed systems. However, as algorithms have gotten rapidly better, recent studies do not engage in consistent, rigorous benchmarking practices, so comparisons across studies are not reliable and cumulative scientific advancement cannot be made.

This paper is a descriptive, quantitative review of 103 exemplary articles on ML-based task scheduling in edge clouds. It is established that there are significant methodological differences: 82% of the studies employ synthetic workloads only, and only 35% perform rigorous benchmarking. Although 95/103 studies claim to use multi-objective optimization, there are extensive utilizations of performance-based metrics (e.g., latency and QoS), whereas reliability and SLA compliance are addressed in less than 12% of studies. In addition, the practices of baseline comparison are not even, and less than 10 % of the studies provide paradigm-balanced comparison.

We perform inferential statistical tests to go beyond descriptive observations and show significant structural dependencies among learning paradigms, dataset types, and benchmarking rigour (chi-square tests, $p < 0.001$, large effect sizes). The findings suggest that the identified differences are not random but systemic, revealing underlying issues in contemporary evaluation practices.

These weaknesses will be addressed by providing a Unified Taxonomy and Benchmarking Framework (UTBF), including a hierarchical method classification and a standardised, reproducibility-aware evaluation protocol. UTBF transforms taxonomy into a working assessment tool, formalizing the strength of benchmarking and cross-paradigm comparisons, and promoting uniform experimental design, unlike traditional survey-based taxonomies. The proposed framework will be used to make fair comparisons, enhance methodological rigour, and perform more deployment-relevant validation in future research on ML-based edge-cloud task scheduling.

Keywords: Edge–Cloud Computing; Task Scheduling; Machine Learning; Benchmarking; Unified Taxonomy; Evaluation Framework.

1. Introduction

The recent rapid spread of latency-sensitive Internet of Things (IoT) applications and services has boosted the uptake of edge-based cloud computing frameworks. Edge computing provides greater privacy, faster responsiveness, and less network congestion than traditional, cloud-centric models by bringing computation closer to data sources [1], [2]. As such, distributed systems are being developed to be even more heterogeneous edge cloud environments.

Task scheduling is one of the major issues of such environments. Unlike conventional cloud data centres, edge infrastructures are described as being heterogeneous, with dynamic loads, limited computational capacity, and high-latency constraints [3]. The process of effective scheduling then has to compromise among conflicting goals, such as latency, energy usage, cost, and resource usage.

To ensure such solutions, machine learning (ML) and deep reinforcement learning (DRL) strategies have become major focuses due to their adaptability to dynamic system conditions [4], [5]. Simultaneously, edge-centric intelligence has recently been studied, with edge nodes partially offloading computation and decision-making to enhance responsiveness and efficiency [1], [2]. Although these advances have enhanced scheduling performance, there have been no correspondingly rigorous and consistent evaluation practices.

Specifically, previous research is characterized by a high degree of diversity in workload choice, benchmarking bottom line, objective modelling, use of the simulator, and reporting on reproducibility. Despite the numerous works that purport to work on multi-objective optimization, latency, and QoS, which are performance-based metrics, they usually prevail in the evaluation. Comparisons across paradigms are not very widespread, and statistical confirmation is used unevenly. Consequently, benchmarking practices remain quite fragmented, which does not allow for fair comparisons and minimizes the external validity of reported results.

To theoretically investigate these concerns, the paper presents a scientific study of 103 articles that analyze the concept of ML-based task scheduling in edge-cloud systems. The analysis does not just cover descriptive categorization, but quantitatively measures benchmarking strength, data utilization, planned diversity, and reproduction practices. According to the specified gaps in the methodology, we develop a Unified Taxonomy and Benchmarking Framework (UTBF) that is a synthesis of hierarchical classification and systematic evaluation process. The proposed framework can contribute to the increased consistency of the comparison and to the methodological rigor of the future studies.

The contributions of this work are principally as follows:

1. A systematic review of 103 edge-cloud ML-based studies of task scheduling.

2. Quantitative determination of the methodological gaps concerning benchmarking practices, data set selection, objective modelling, and the use of the simulator.

3. A Unified Taxonomy and Benchmarking Framework (UTBF), a compilation of classification and a standardized evaluation point of view.

4. Recommendations on practical guidelines to aid reproducibility, cross-paradigm comparison, as well as better experimental design in further studies.

The rest of this paper is structured as follows. Section 2 is a review of related work. The methodology, together with the construction of the data set, is presented in Section 3. In section 4, the results of the analysis are provided. Research gaps are discussed in Section 5. The proposed framework is presented in Section 6. Threats to validity are described in section 7. Section 8 concludes the paper.

2. Related Work

In distributed systems, research on task scheduling has advanced significantly with the emergence of edge-cloud computing. The first scheduling methods available in traditional cloud environments were largely heuristic and rule-based, such as First-Come-First-Served (FCFS), Min-Min, and Round Robin [7], [8]. These techniques have been developed in relatively stable, homogeneous environments, whereas edge cloud environments are more heterogeneous, with fluctuating workloads, fewer computational resources, and more rigid latency constraints. Consequently, further adaptive scheduling plans have become more extensively investigated.

2.1 Metaheuristic-Based Scheduling

The PSO, Genetic Algorithms (GA), and Ant Colony Optimization (ACO) are metaheuristic algorithms that have been extensively used to solve distributed task scheduling problems due to their ability to search a vast solution space under a variety of constraints [9], [10]. These approaches are often able to generate superior approximate solutions in complex optimization contexts compared to deterministic schedulers. To a significant extent, however, most of this literature relies on synthetic simulation scenarios, and performance is measured against a relatively small set of baselines. Moreover, comparisons of fundamentally different scheduling paradigms, including learning ones, are uncommon in most studies.

2.2 Machine Learning and Deep Reinforcement Learning Approaches

The shortcomings of manual scheduling policies have given rise to growing interest in machine learning (ML) and deep reinforcement learning (DRL) techniques. Workload prediction, classification, and decision support in scheduling systems have been addressed using supervised learning [11], but DRL has increasingly been used for adaptive resource management in uncertain, non-stationary scenarios [4] and [12]. Deep Q-Networks (DQN) and policy-gradient methods, such as actor-critic methods, have demonstrated encouraging performance in latency-constrained edge scenarios [12], [13].

Meanwhile, these approaches provide trade-offs. Deep models can enhance adaptive decision-making; however, they increase training complexity, memory requirements, and computational cost, which can be considerable in resource-limited edge environments [6]. This places the evaluation design in a special position of importance, as any reported gains cannot be evaluated based on performance metrics alone; they should be evaluated in relation to deployment feasibility, model cost, and the experimental conditions.

2.3 Edge Intelligence and Distributed Processing Frameworks

In addition to time-scheduling research, edge intelligence has received considerable attention, in which preprocessing, inference, and decision-making are moved closer to data sources [1], [2]. Such architectures can reduce latency, enhance responsiveness, and reduce reliance on centralized cloud resources. They are also more effective at supporting privacy-sensitive, bandwidth-constrained applications than an entirely cloud-based solution.

Irrespective of these architectural developments, assessment practices are inconsistently assessed across studies. Numerous publications focus on algorithmic or system-level novelty, whereas the workload is characterized in a vastly different fashion, the simulator is configured, reporting is detailed, and validation processes are conducted differently. That is why one cannot easily compare the reported results across studies, even when they concern similar scheduling objectives.

2.4 Existing Taxonomies and Survey Studies

Some of the surveys and taxonomy research have examined scheduling techniques in both cloud and edge settings [14], [15]. Such works typically divide the literature by algorithm families, system architectures, or application scenarios. These classifications can be used to organize the field and identify significant directions for research.

But earlier surveys have been mostly descriptive in their nature. They do not make clear evaluation dimensions for benchmarking strength, dataset diversity, reproducibility, or cross-paradigm fairness in most instances. Consequently, they offer little assistance in determining the quality or comparability of methodology in the available research. This necessitates a common point of view that not only classifies scheduling approaches but also enables a more systematic and consistent review. In contrast to previous surveys, the study directly defines benchmarking strength as an evaluative dimension (formally) and combines taxonomy with a benchmarking protocol that is reproducible.

3. Research Methodology

The paper employs a data-driven review approach to systematically examine trends in methodology, benchmarking practices, and objective modelling strategies in machine-learning-based task-scheduling studies in edge-cloud settings. Instead of the narrative comparison, the adopted approach turns the reviewed literature into a clear-cut dataset that will support quantitative analysis and cross-study comparison.

Fig. 1 shows the entire research process, including study selection, manual annotation, validation, and analysis.

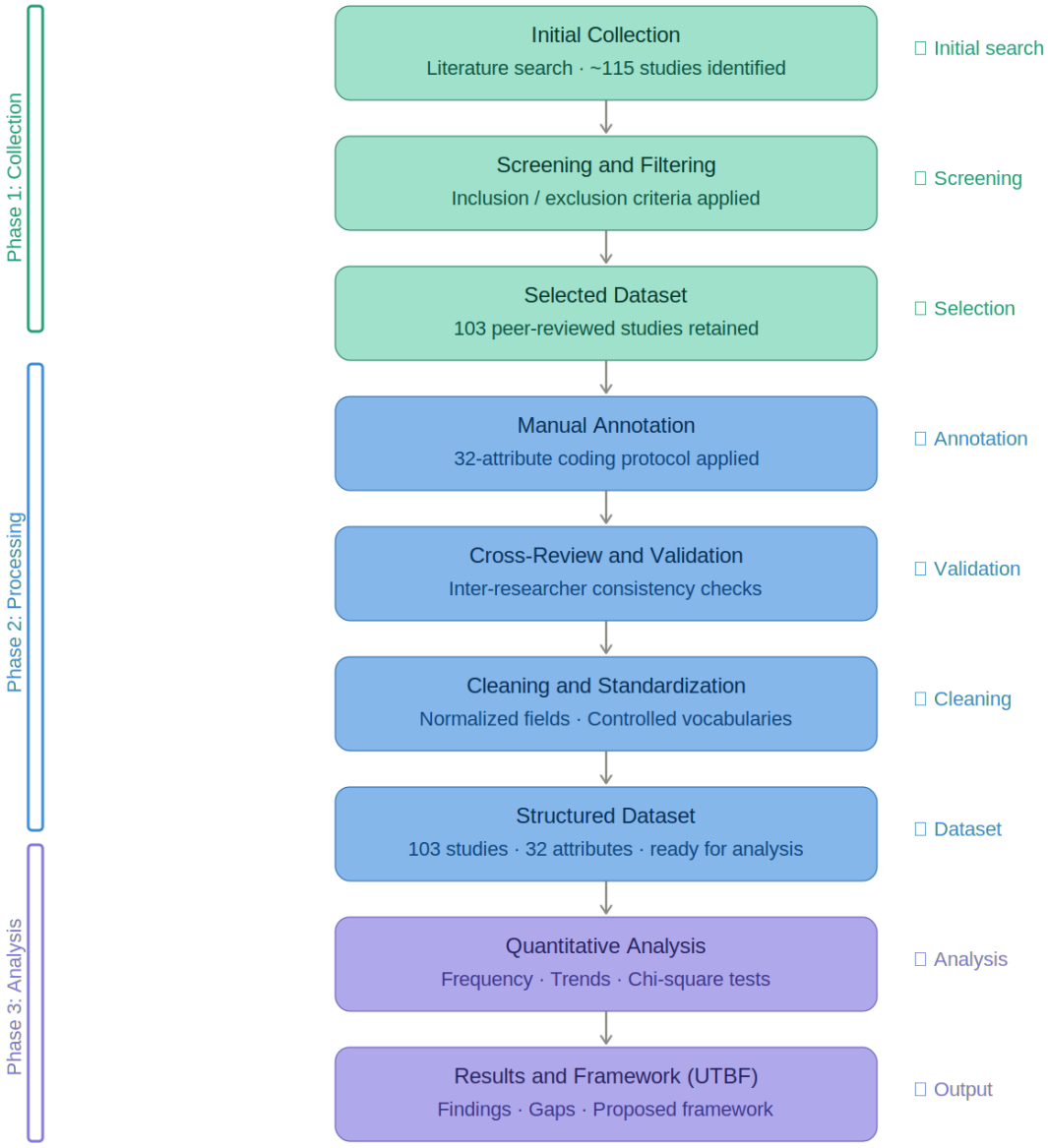

Fig. 1. Research workflow illustrating the study selection, annotation, and analysis pipeline adopted in this study

3.1 Study Collection and Selection Process

The study corpus was formed through a multi-stage selection process. A preliminary list of 115 records was compiled from the literature on task scheduling, task offloading, and resource allocation in edge-cloud and mobile edge computing settings, obtained through targeted searches of key academic databases using combinations of keywords.

A sieving step was then performed based on predetermined inclusion and exclusion criteria. To be included in the research, the studies had to:

(1) dealt with scheduling or offloading tasks or resources in edge-cloud, MEC, or similar distributed systems;

(2) utilized machine learning, reinforcement learning, metaheuristic, or hybrid optimization algorithm.

(3) provided experimental evaluation results

(4) have been published either in peer-reviewed journals or accepted conference locations.

Literature that focused entirely on theoretical representation without experimental verification was excluded. Moreover, papers whose main theme was not scheduling (e.g., network routing) or that lacked an edge-cloud background were excluded.

After this process, 103 studies were left to be structured, coded, and quantified.

3.2 Manual Annotation and Coding Protocol

To ensure uniformity in data typing, a coding protocol was created before annotation and applied uniformly across all study groups. The protocol includes seven coding manuals in each category, whose attributes are defined, permissible values are specified, and explicit decision rules are provided, all of which are specific to a particular methodological paradigm, along with inclusion and exclusion criteria. These manuals are supplementary materials that can promote transparency and replicability in the annotation process.

One researcher annotated all the studies, and another reviewed them independently. More systematic observation of arguments was also made, and their resolution was based on discussion through the established coding protocol, not on intuition. This was done until all the entries agreed on the general response. Though no formal inter-annotator agreement measures were conducted (e.g., Cohen's Kappa), a structured procedure that combines predefined decision rules, independent review, and consensus-based resolution was adopted to ensure coding consistency, as is customary in other data-driven survey studies.

The annotation was also intended to reflect the methodological characteristics, not the quality of specific studies. The coding decisions were therefore made solely on what each paper actually said, and no attribute was ever decided based on context or convention.

3.3 Data Extraction and Structured Encoding

The selected works were coded using the structured taxonomy template, which included 32 attributes describing conceptual, methodological, and evaluative issues. These are learning paradigms, families of algorithms, objective formulations, types of data to use, evaluation settings, baseline comparison strategies, strengths of benchmarking, and pointers to reproducibility.

These data have been obtained mainly from the methods and assessment sections of the articles and matched to the standardized values using controlled vocabularies. By doing so, there will be fewer variations in wording, making it more feasible to achieve a consistent aggregation of studies.

Later in the processing stage, the attributes were selected, cleaned, and recoded, and the resulting normalized fields were prepared for quantitative analysis.

3.4 Benchmarking Strength Classification

To make the evaluation rigorous and comparable, the benchmarking practices were divided into three ratings: weak, moderate, and strong. The classification was made based on rule-based criteria based on observable and expressly reported characteristics of each study, and in line with the decision rules outlined in the coding manuals.

In strong benchmarking, cross-paradigm baseline comparisons (i.e., comparisons with methods from at least two different methodological paradigms) are required, as are statistical validation (e.g., confidence intervals or significance testing) and adequate disclosure of reproducibility (e.g., simulator identification, parameter reporting, and workload description). Moderate benchmarking is a set of comparisons made between the bases, but it is within a single methodological paradigm, with minimal statistical reporting and only limited information on reproducibility. Weak benchmarking is defined by one or intra-paradigm basis comparisons, lack of statistical validation, and little to no data on reproducibility.

The reason for this formalization is twofold. First, it enables benchmarking strength to be evaluated as a comparable dimension, measurable across studies, rather than as a description. Second, it guarantees that the classification as a reflection of the nature of structural evaluation - the diversity of comparison methods and the completeness of reportage of the experiment - is not a subjective evaluation of the quality of research. This is equally true of all the methodological paradigms in the dataset.

3.5 Quantitative Analysis Procedure

The coded data were interpreted through structured quantitative approaches. It will be analyzed using frequency distributions across the dimensions of taxonomy, trends over time, cross-dimensional interactions (e.g., between learning paradigms and dataset types), baseline comparison patterns, and objective diversity evaluation.

Programmatic creation of all analytic outputs, such as figures and summary tables, was performed within a controlled computing workflow, and traceability between the encoded dataset and the reported results was ensured.

3.6 Validity and Reproducibility Measures

A few measures were taken to increase the study's internal consistency and analytical reliability. Standardized vocabulary categories were used for encoding, and compatibility checks of similar

attributes were performed to identify any missing or contradictory records. Attributes that were not identifiable using the reported content were not inferred and were merely left unencoded.

The encoded dataset and analysis scripts are published alongside the predetermined coding manuals, detailing decision rules and permissible values for each category, in a reproducibility package. The table and summary tables used in this article are the results of programmatic computation of structured data using repeatable Python scripts, and both the encoded data and the reported results are fully traceable. It is a package provided to enable independent checking and extend the analytical pipeline.

All of the above would contribute to the transparency and reproducibility of the methodological pipeline employed in this research by assisting in ensuring that the application of category-specific coding manuals, cross-reviewing (independent), consensus-based resolution of conflicts, controlled vocabularies, and programmatic analysis are facilitative.

4. Results and Analytical Insights

Section 4 gives a quantitative analysis of the 103 peer-reviewed articles that are contained in the curated dataset [16]-[118]. These are the analytical articles to serve as the analytical corpus of the proposed research and were chosen based on the process described in Section 3.

The analysis is conducted in an aggregate manner where standardized attributes are determined and contrasted in the studies rather than providing separate descriptive summaries. The discussion covers various points, which include, learning paradigm, objective modelling techniques, use of datasets, benchmarking and evaluation environments.

The whole system of statistical distributions, temporal trends, and cross-dimensional correlations, which are introduced in this section, is founded on the complete dataset. This ensures that there is uniformity in interpretation and traceability between the encoded data and the reported findings.

4.1 Evolution of Learning Paradigms

The dynamic nature of learning paradigms in ML-based task scheduling for edge-cloud environments has undergone significant change in recent years.

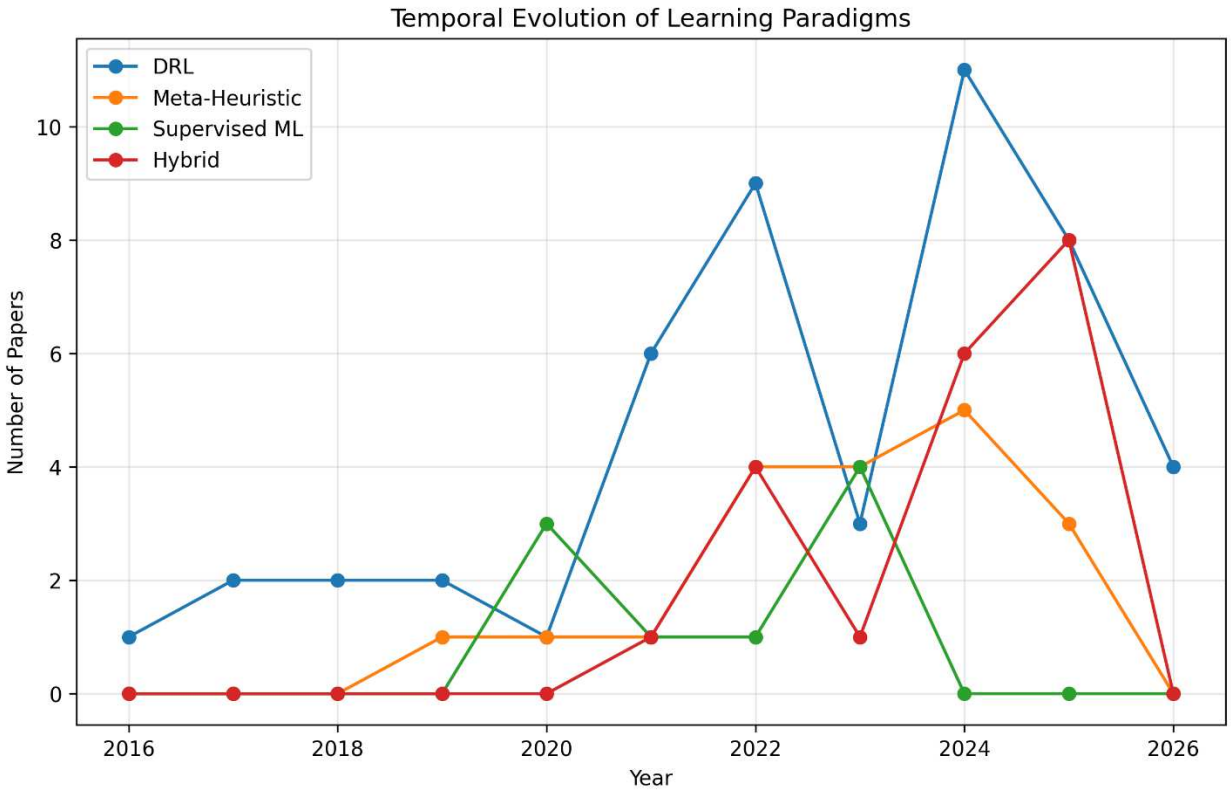

Figure 2: Temporal Evolution of Learning Paradigms in ML-based Edge–Cloud Task Scheduling

The DRL-based approaches have increased dramatically since 2021, as shown in Figure 2, and have become increasingly popular in recent research. Hybrid approaches, especially those integrating DRA with a metaheuristic or assistive ML elements, have demonstrated the fastest growth, most notably between 2022 and 2025. By contrast, classical metaheuristic methods do not show such a pronounced increase in adoption.

This trend represents a fundamental shift towards standalone heuristic optimization, learning-based scheduling strategies, and hybrid approaches. Meanwhile, the growing variety of learning paradigms underscores the need for stable, comparable benchmarking practices, as these paradigms can be tested under heterogeneous experimental conditions.

4.2 Benchmarking Rigour Over Time

The rigour of benchmarking was studied within the framework of learning paradigms to understand differences in evaluation methodology.

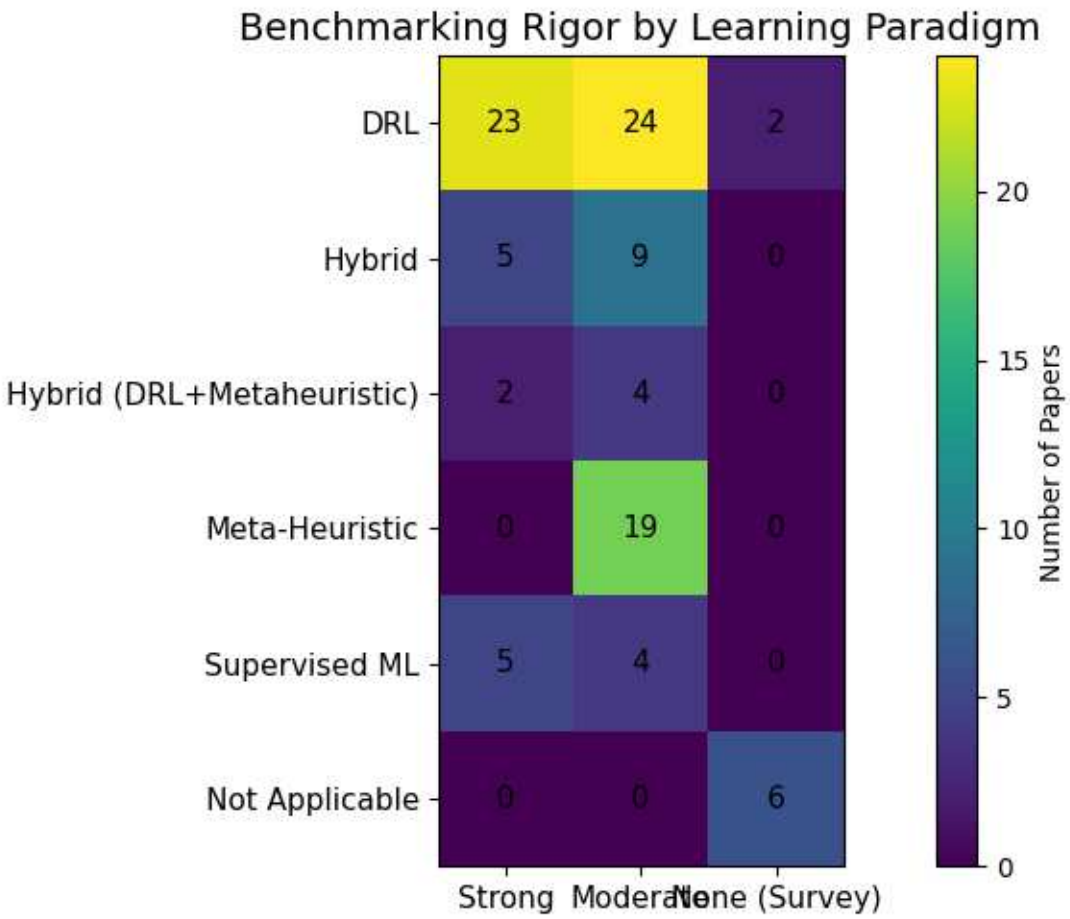

Figure 3: Benchmarking Rigour by Learning Paradigm

The benchmarking levels are distributed among various learning paradigms as shown in Figure 3. Studies based on DRL can be split into strong and moderate benchmarking, and a few into the survey-only category. Hybrid methods continue to focus primarily on the moderate category, whereas metaheuristic research is linked to nearly all moderate benchmarks. The trends of supervised ML studies indicate that the distributions of strong and moderate benchmarking are more balanced, whereas survey papers are inherently assigned to the non-experimental category.

This trend indicates that benchmarking rigour is not uniformly distributed across methodological paradigms. Otherwise, paradigms are more likely to be linked to different sets of evaluation practices, implying that benchmarking decisions can be based on the adopted methodology rather than on a field-wide standard.

As a further test of whether the rigour of benchmarking varies systematically with methodological characteristics, two chi-square tests of independence were conducted.

To begin with, the connection between the learning paradigm and the level of benchmarking was examined. The findings reveal that the relationship is statistically significant ($\chi^2 = 92.82$, df = 10, p = 0.001), and the effect size is large (Cramér's V = 0.67; N = 103). This implies that the practice of benchmarking does not occur randomly in the selection of a learning paradigm, but instead has a systematic connection.

Second, the correlation between dataset type and the level of benchmarking was considered. The chi-square test also demonstrated a significant relationship ($\chi^2 = 97.95$, 8 df, p = 0.001), and the findings are high. It means that research based on synthetic workloads is more often associated with lower benchmarking rigour than research using real-world datasets.

When put together, these results provide quantitative evidence that the differences in methodology across the literature are structural rather than accidental. They also support the idea that more consistent, similar benchmarking practices should be used across more studies.

4.3 Baseline Comparison Practices

We next examined baseline comparison strategies at the per-paper level.

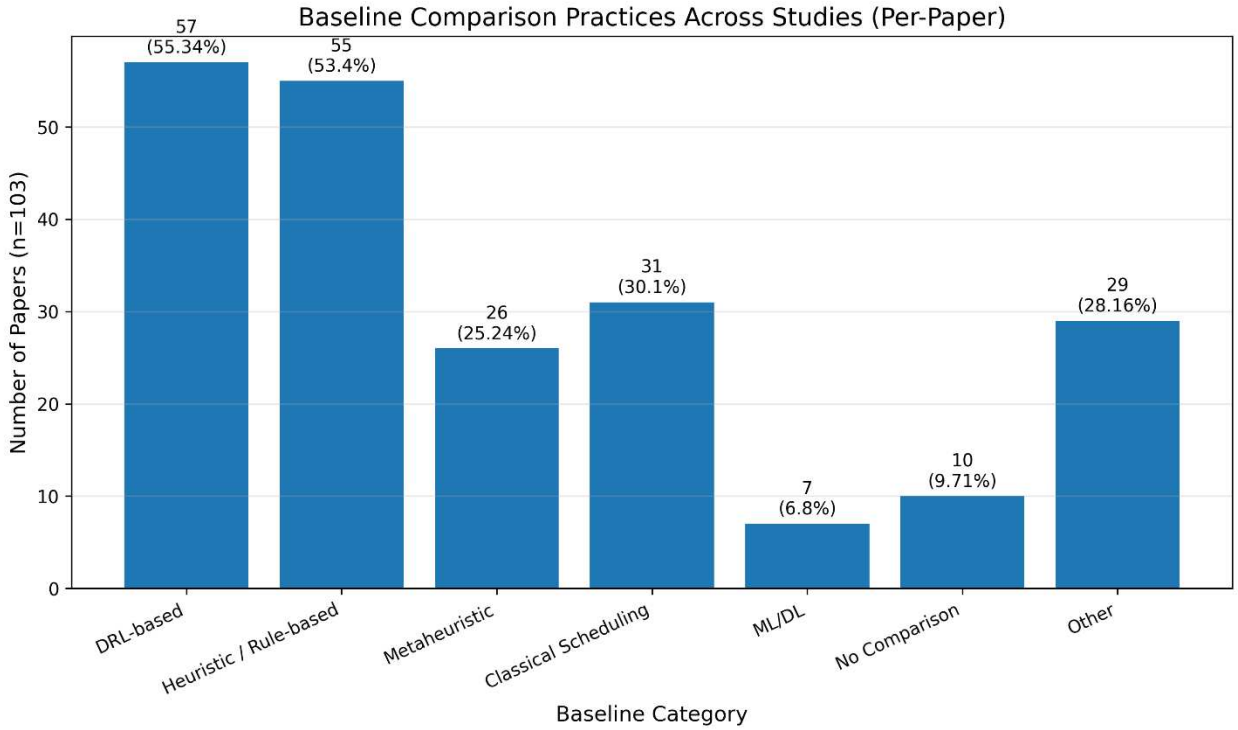

Figure 4: Baseline Comparison Practices Across Studies (Per-Paper)

Figure 4 indicates that 57 studies (55.34%) compare to DRL-based baselines, and 55 studies (53.4%) use heuristic or rule-based baselines. However, by comparison, classical scheduling baselines (e.g., FCFS and Round Robin) are included in only 31 studies (30.1%) and 7 studies (6.8%), respectively, compared to ML/DL-based methods. These %ages do not sum to 100% because individual studies may include multiple baseline categories.

All in all, this distribution indicates a propensity towards intra-paradigm or selectively chosen comparisons, in which studies tend to benchmark their methods against those of the same paradigm. Such practice can decrease cross-paradigm fairness and narrow down the scope of evaluation. This trend can be further demonstrated by the temporal development of the baseline practices.

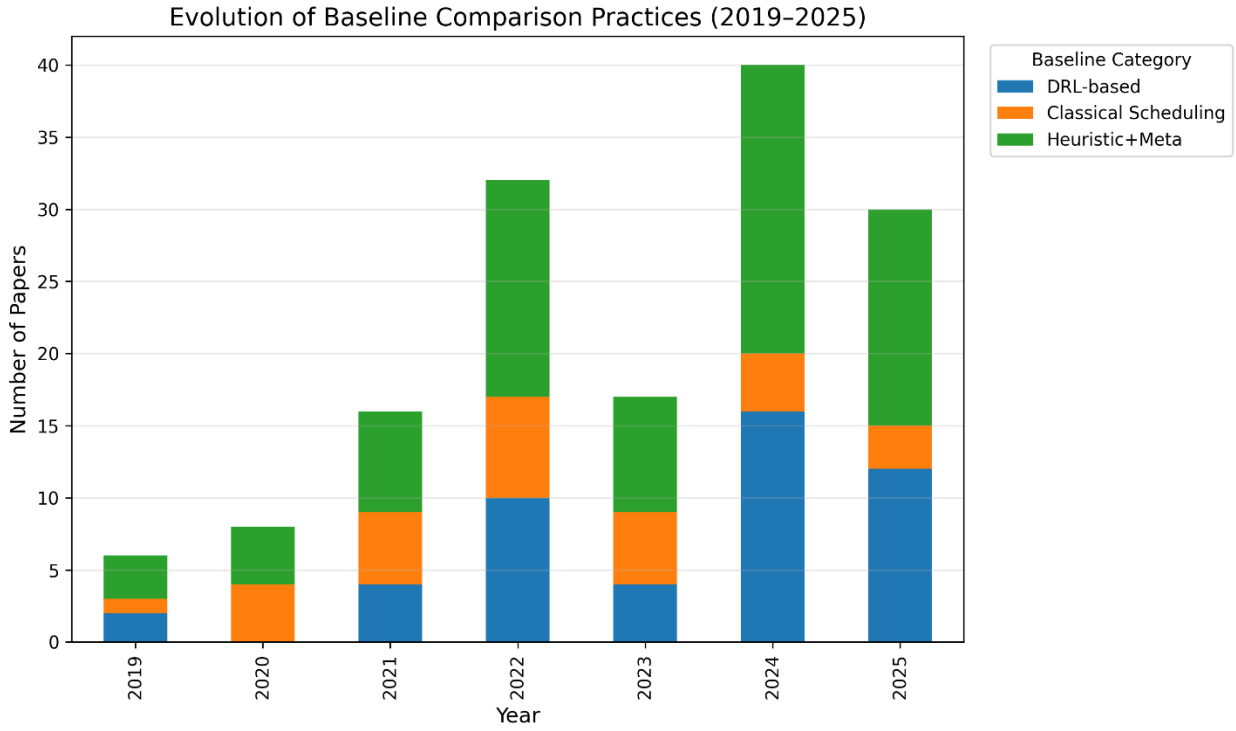

Figure 5: Evolution of Baseline Comparison Practices

As Figure 5 indicates, comparisons relying on DRL grow significantly after 2021, as the use of learning-based scheduling methods continues to grow. Meanwhile, the comparative application of classical scheduling baselines decreases. Another widely used technique, year after year, is the use of heuristic and metaheuristic baselines, which are often presented alongside learning-based approaches.

The numbers in Figure 5 do not sum to zero, since individual studies can have multiple baseline categories. All in all, the trends show that benchmarking practices have shifted to include learning-based references and to reduce reliance on foundational scheduling baselines.

4.4 Dataset Usage and External Validity

One of the most notable findings concerns dataset selection.

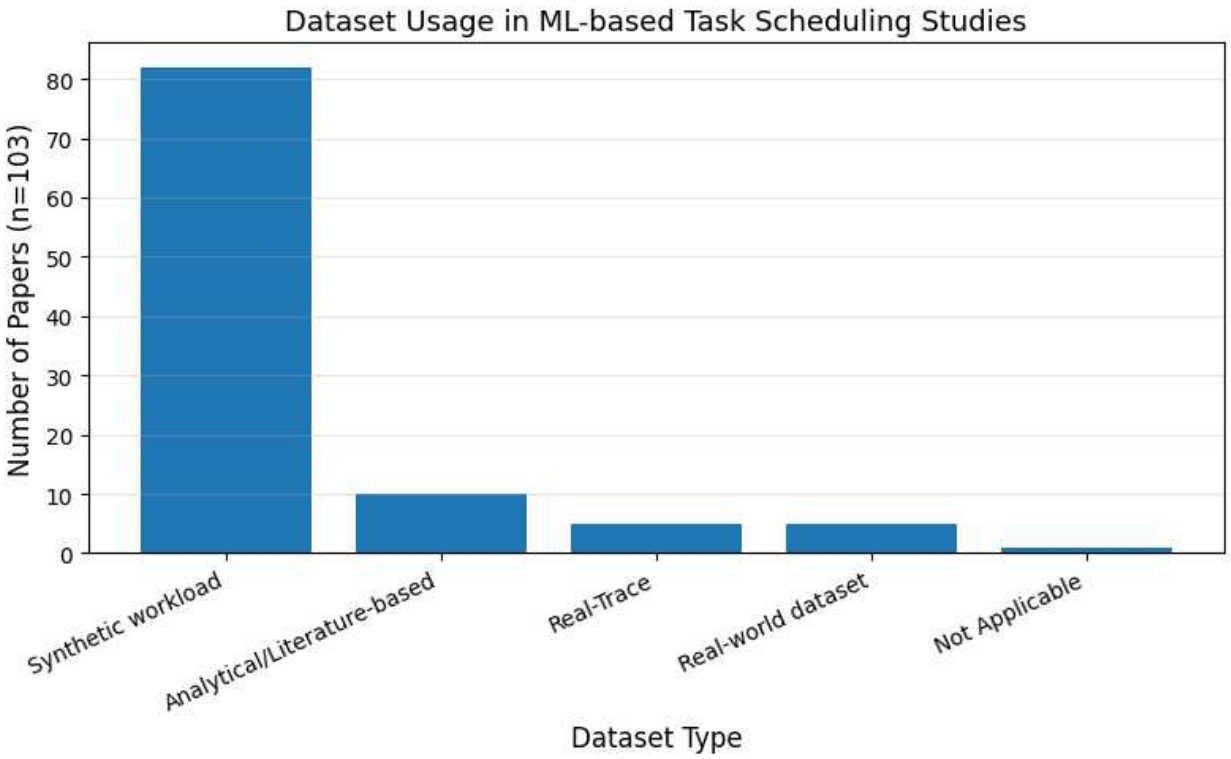

Figure 6: Dataset Usage Distribution

Table 1 and Figure 6, presented in summary form, show that 82 studies are based on synthetic workloads (79.61%). Conversely, real-world data and trace-based assessments collectively amount to 10 studies (9.71) in the analytical corpus.

Further cross-analysis shows that advanced paradigms, in particular, DRL, hybrid, and metaheuristic, rely largely on synthetic data.

Table 1: Dataset Type by Learning Paradigm

Learning Paradigm	Synthetic	Analytical	Real-Trace	Real-World	Total
DRL	45	3	0	1	49
Meta-Heuristic	19	0	0	0	19
Supervised ML	5	5	3	2	15
Hybrid	7	1	1	0	9
Hybrid (DRL+Meta)	6	1	1	0	8
Other	—	—	—	—	—

Precisely, over 90% of studies based on DRA use synthetic workloads, and all studies based on metaheuristics in the dataset rely solely on synthetic data. Quite to the contrary, real-world data appear to be more prevalent in supervised learning investigations than in other paradigms.

This extensive use of synthetic data poses substantial problems of external validity, deployment realism, and the generalizability of reported scheduling performance. This extreme reliance on synthetic data, with no corresponding analysis of real-world datasets or trace-based workloads, is a concern for external validity and deployment realities.

4.5 Objective Function Diversity

Even though a high percentage of the studies reviewed are described as multi-objective, the distribution of the optimization targets is skewed.

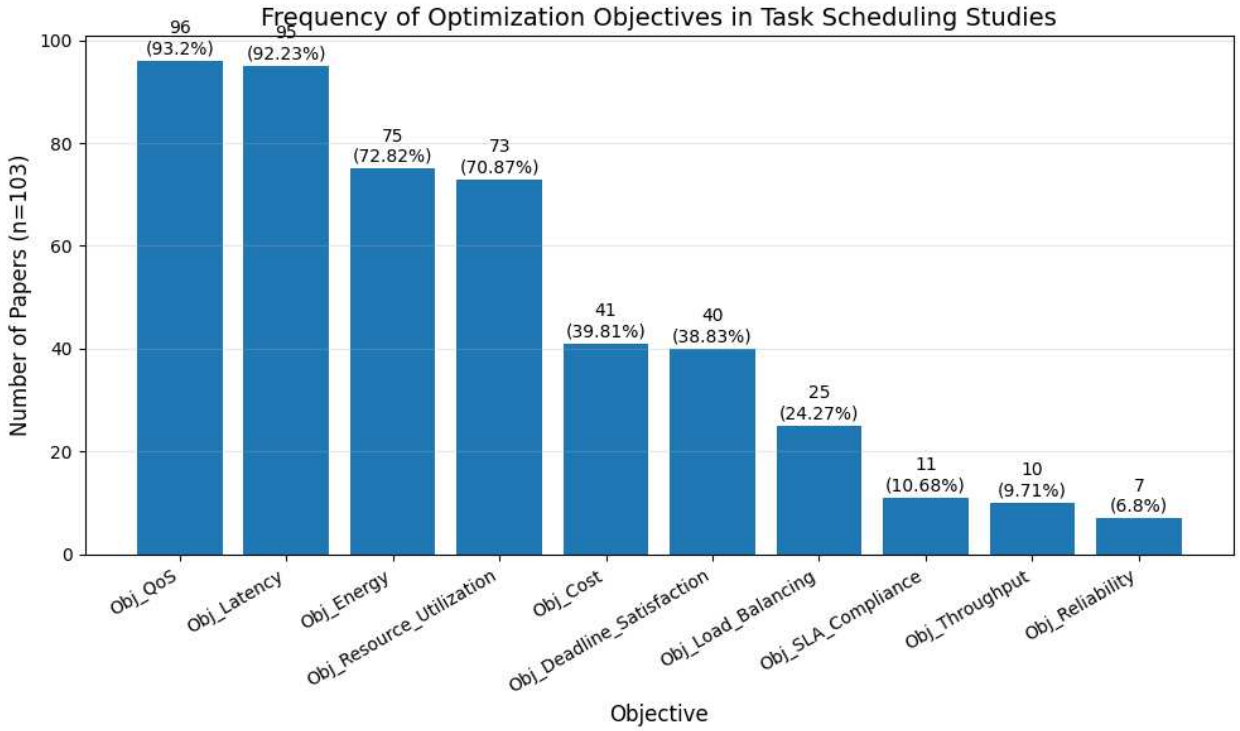

Figure 7: Objective Frequency Distribution

As Figure 7 shows, the most common objectives to maximize are latency and QoS, which are applied in 95 (92.23) and 96 (93.2) studies, respectively. These are then preceded by energy consumption (75 studies, 72.82) and resource utilization (73 studies, 70.87), which are also commonly found in scheduling optimization models.

Quite to the contrary, several goals are not represented significantly. Compliance with SLA (11 studies, 10.68) is only 11, throughput (10 studies, 9.71) is only 10, and reliability (7 studies, 6.8) is only 7. Other goals, such as cost satisfaction and meeting deadlines, are considered at a moderate level, but not to the same extent as performance-based measures.

The analysis reveals that in spite of the wide usage of multi-objective formulations, the scope of optimization is still limited to a few objectives in practice. In particular, the emphasis on latency

and QoS means that performance-based evaluation is more significant, while aspects related to service guarantees, system robustness, and long-term operational availability are less emphasized.

Therefore, current models of multi-objective scheduling might not fully capture the intricacies of real-world deployment requirements, and a more balanced, all-encompassing modelling of the objectives should be a research agenda for the future.

4.6 Simulator Fragmentation and Reproducibility

Lastly, benchmarking contexts were analyzed to evaluate the degree of fragmentation of experiments in the literature.

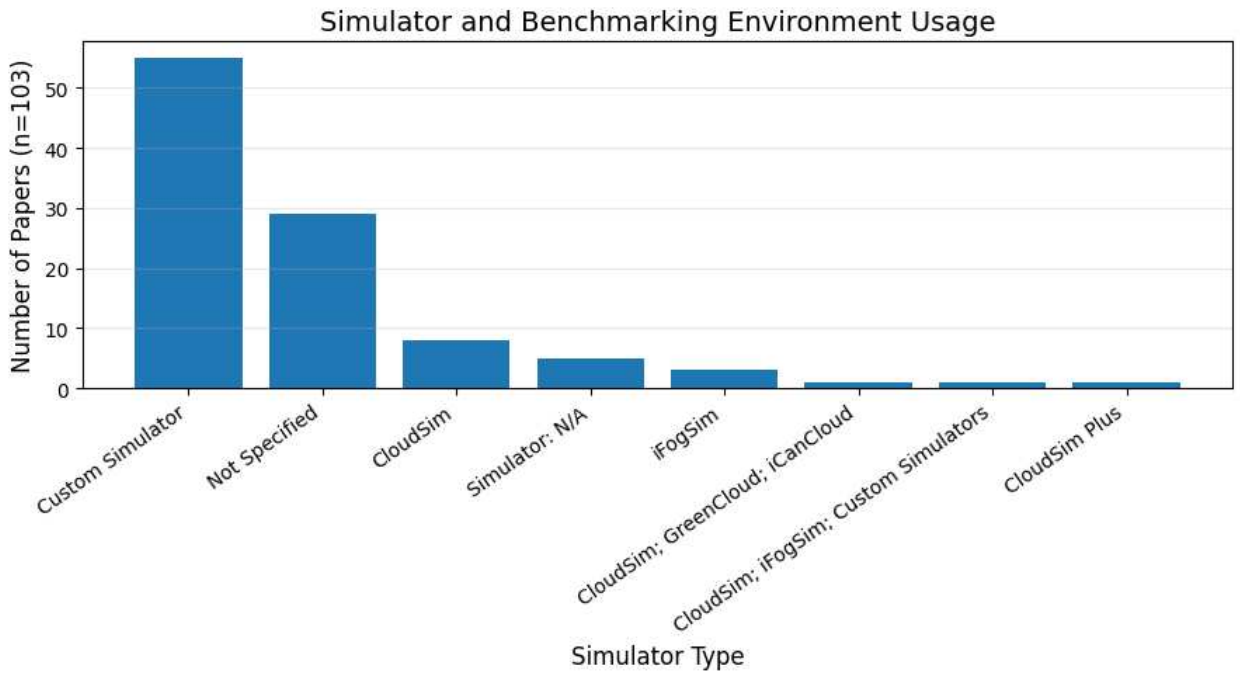

Figure 8: Simulator and Benchmarking Environment Usage

Table 2. Simulator and Benchmarking Environment Usage

Simulator / Environment	Number of Studies	%age (%)
Custom Simulator	55	53.4%
Not Specified	29	28.16%
CloudSim	8	7.77%
iFogSim	3	2.91%
Other Simulators	8	7.77%
Total	**103**	**100%**

Figure 8 and Table 2 show that more than half of the studies (55 papers, 53.4) employ custom-built simulators, and 29 studies (28.16) do not mention their evaluation platform. In comparison, popular simulation tools such as CloudSim and iFogSim can be found in 8 and 3 studies, respectively, and make up 10.68% of the articles analyzed collectively.

The degree of fragmentation of the benchmarking practices is reflected in the dependence on custom or unspecified environments. Such non-standardization restricts reproducibility and cross-study comparisons since results are generally specific to particular non-shared simulations.All in all, the observed inconsistency in the use of the simulator indicates that measurement settings are not aligned across studies, supporting the need for a standardized, consistent benchmarking framework.

5. Research Gaps and Future Directions

The analytical results presented in Section 4 indicate common methodological gaps in the existing literature on ML-based task scheduling in edge clouds. Taken together, these gaps demonstrate that a more systematic and consistent design and benchmarking of taxonomies is needed.

5.1 External Validity Gap: Dependence on Synthetic Workloads

The review indicates that a significant %age of the articles studied are based on synthetic workloads, and the %age that use real-world datasets or trace-based analyses is low. State-of-the-art paradigms, especially DRL and hybrid models, are also largely reliant on synthetic data.

The statistical outcomes in Section 4 also support this imbalance. The strong correlation between the type of dataset used and the degree of benchmarking suggests that the use of synthetic workloads is associated with poorer benchmarking practices, rather than being randomly distributed across studies.

Synthetic data can be handy for controlled experimentation, but excessive use of it without additional analysis of real-world data sets or trace-based workloads can undermine external validity, restrict deployment applicability, and limit cross-study comparability.

Future directions are the creation of open benchmark trace repositories of edge-cloud systems, more explicit workload-generation practices, and more extensive use of trace-driven or real-data-driven validation, as well as simulation.

5.2 Benchmarking Rigour Gap: Paradigm-Dependent Evaluation Practices

The findings also indicate that the rigour of benchmarking varies across methodological paradigms. Certain categories are more often associated with robust evaluation activities, while others remain confined to moderate benchmarking environments. Moreover, comparisons of baselines are often intra-paradigm, thus limiting the scope and impartiality of comparative assessment.

The latter differences are supported by the statistical analysis in Section 4, which found that the level of benchmarking and the learning paradigm were significantly related. This implies that practice in evaluation is influenced, at least in part, by methodological tradition rather than by a standard of benchmarking used throughout the discipline.

Therefore, the literature cannot be easily compared systematically across paradigms, despite studies discussing similar scheduling problems.

The way forward is to embrace clearer benchmarking processes, wider application of cross-paradigm baselines, and more uniform reporting of reproducibility and statistical validation procedures.

5.3 Objective Diversity Gap: Concentration on Performance-Centric Metrics

Most studies are stated as multi-objective, but the objectives optimized focus on a relatively limited set of performance-related indicators, in particular, latency and QoS. Reliability, SLA compliance, and throughput seem to be far less often in the list of objectives.

This concentration shows that the edge cloud scheduling is only partially modeled in the current optimization practice. In particular, it may be underexpressed in aspects such as service guarantees, resilience, and the system's long-term behavior.

Future directions include greater use of reliability-sensitive and SLA-sensitive goal models, more balanced multidimensional optimization schemes, and greater emphasis on goal-related aspects of robustness, fairness, and operational stability.

5.4 Reproducibility Gap: Simulator Fragmentation and Limited Standardization

The other gap concerns the disintegration of experimental settings. A large portion of the literature is based on custom simulators, and many other studies do not specify the evaluation platform. In comparison, the most popular tools, including CloudSim and iFogSim, account for only a relatively small portion of the literature.

This inconsistency makes it difficult to replicate the reported results and even to compare studies conducted under similar experimental conditions. Not all of the time is the problem; the issue is the absence of standardization of reporting and shared evaluation environments throughout the field.

New directions include standardizing simulator reporting practices, publishing code and experimental settings, and community-supported benchmarking.

5.5 Toward a Unified Taxonomy and Benchmarking Framework

The observed imbalance in datasets, variations in benchmarking due to paradigm dependencies, objective concentration, and fragmentation of simulators indicate that there is no sufficiently consistent evaluation ecosystem to support research on ML-based edge-cloud task scheduling.

Section 4 also presents statistical results that again suggest these inequalities are not by chance. This creates an evident motivation towards a standard framework that can be applied to make a more systematic comparison of studies.

It is believed that a framework of this sort acts as a guide towards improving cross-paradigm comparability, methodological clarity, reproducibility, and evaluation transparency. It is not intended to establish a general protocol to be followed by all studies, but rather a flexible guideline to provide a more balanced and understandable benchmarking practice.

The following section introduces the proposed Unified Taxonomy and Benchmarking Framework (UTBF), which is aimed at addressing these gaps with a systematic, sensitivity to reproducibility, and evaluation procedure

6. Proposed Unified Taxonomy and Benchmarking Framework

This section presents a Unified Taxonomy and Benchmarking Framework (UTBF) for ML-based task scheduling in edge-to-cloud computing systems, addressing the methodological gaps identified in Section 5. The framework will be applied to support more consistent classification and evaluation practices through fragmentation in benchmarking, inconsistency in objectivity modeling, and a lack of standardization of experimental reporting

The UTBF motivation is further explained by the statistical results of Section 4. Specifically, the chi-square tests show that there are considerable relationships between learning paradigms and benchmarking rigour, between dataset types and benchmarking strength ($p < 0.001$; large effect sizes). These findings indicate that differences in methodology across the literature are systematic rather than isolated, providing an excellent foundation for a more organized, cross-paradigm evaluation.

The proposed framework is not intended to be a mandatory protocol but rather a reference structure to enhance comparability, methodological transparency, and reproducibility across studies. The framework can be divided into two complementary components:

1. A hierarchical taxonomy layer.

2. A uniform benchmarking protocol layer.

Table 3 compares UTBF with six exemplary studies across seven evaluation dimensions to situate the proposed framework within the current body of literature. The studies were chosen to represent the variety of taxonomic and benchmarking methods already in use in the field, such as algorithm-based taxonomies, surveys of performance metrics, and domain-specific surveys. The comparison is structured around the dimensions that are core to the framework design paradigm classification, benchmarking formalization, cross-paradigm baseline requirements, dataset diversity assessment, statistical validation, reproducibility reporting, and the existence of a structured evaluation protocol. As illustrated in Table 3, the current literature covers these dimensions either partially or on a small scale. Houssein et al. present a systematic taxonomy and only extend it to metaheuristic techniques. Aslanpour et al. address performance measures and evaluation criteria, but the strength of benchmarking is not formalized as a classification criterion used across all studies. The remaining surveys incompletely classify the paradigm and fail to cover other dimensions. Collectively, this comparison demonstrates that no existing work combines all seven dimensions into a single, operationally feasible framework, thereby stimulating the conceptualization of UTBF.

Table 3: Dimensional comparison of existing surveys and taxonomy studies against the proposed Unified Taxonomy and Benchmarking Framework (UTBF).

Study	Hierarchical paradigm classification	Benchmarking strength formalized	Cross-paradigm baseline required	Dataset diversity assessed	Statistical validation included	Reproducibility as an explicit dimension	Structured evaluation protocol	Coverage
Houssein et al. (2021) Meta-heuristic taxonomy, cloud scheduling	✓	—	—	—	—	—	—	1/7
Aslanpour et al. (2020)	◑	◑	—	◑	◑	—	—	4◑/7

Study	Hierarchical paradigm classification	Benchmarking strength formalized	Cross-paradigm baseline required	Dataset diversity assessed	Statistical validation included	Reproducibility as an explicit dimension	Structured evaluation protocol	Coverage
Performance evaluation metrics, benchmarks & standards								
Dong et al. (2024) Task offloading strategies survey, MEC	◑	—	—	◑	—	—	—	2◑/7
Ismail et al. (2025) DRL-based scheduling survey, MEC	◑	—	—	—	—	—	—	1◑/7
Islam et al. (2022) Edge-cloud scheduling survey, issues & challenges	◑	—	—	—	—	—	—	1◑/7
Singh et al. (2025) Task scheduling survey, edge-cloud environments	◑	—	—	◑	—	—	—	2◑/7
UTBF (This work) Data-driven unified taxonomy & benchmarking framework	✓	✓	✓	✓	✓	✓	✓	**7/7**

✓ = explicitly addressed as a formal dimension; ◑ = mentioned but not formalized or systematically applied; — = outside the stated scope of the study.

6.1 Unified Taxonomy Layer

The taxonomy element of the suggested framework offers a hierarchical categorization of studies on ML-based task scheduling across five fundamental dimensions that reflect the different facets of methodological design and critique. These dimensions constitute the first phase of the UTBF application workflow, as shown in Phase 1 of Figure 9, in which a study is organized before the benchmarking protocol.

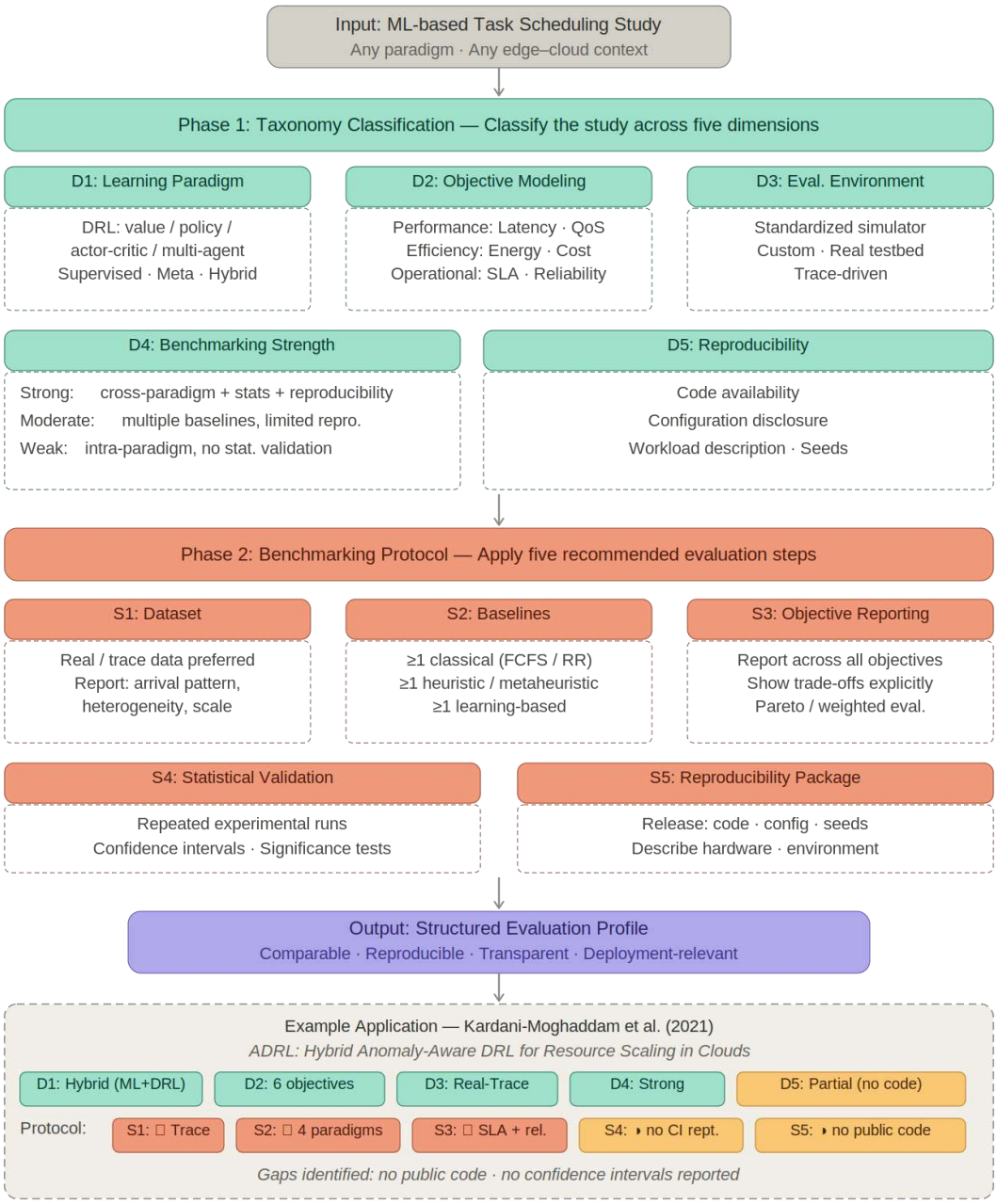

Figure 9: Proposed UTBF architecture, the taxonomy layer and the benchmarking protocol layer.

The first dimension is the learning paradigm, which divides studies by the methodological approach they use. It encompasses deep reinforcement learning (DRL) algorithms, including value-based, policy-based, actor-critic, and multi-agent variants; supervised or predictive machine learning models; metaheuristic optimization methods; and combinations of multiple paradigms. The explicit classification of these categories facilitates cross-paradigm comparability and minimizes siloed evaluation practices, which, as found in the statistical analysis of Section 4.2, were identified as a systemic feature of the existing body of literature.

The second dimension is the objective modeling, which transcends the classical duality of single- and multi-objective models. Instead, the framework directly represents the type of optimization objectives to achieve in any study in three groups: performance-driven (latency, QoS, throughput) objectives, efficiency-driven (energy, cost, resource utilization) objectives, and operational (reliability, SLA compliance, load balancing) objectives. This kind of systematic organization, as in Figure 9, aids in clarifying objective selection and reducing multi-objective ambiguity in broad labeling, the issue of which is that reliability and SLA compliance are reported in less than 12% of the sources reviewed, whereas all sources assert them.

The third dimension addresses the evaluation environment and is intended to address the piecemeal benchmarking practices identified in Section 4.6. Research falls into the categories of using standardized simulators (e.g., CloudSim or iFogSim), custom-made simulators, testbeds (real world), or trace-driven evaluations. In this dimension, the framework facilitates the reporting of important configuration parameters, thereby making interpretation of results easier and enabling comparisons across studies.

The fourth dimension standardizes benchmarking power as an evaluative characteristic that can be measured rather than described. The rigor of benchmarking, depicted in Figure 9, is divided into three levels with clear criteria: strong benchmarking, which entails cross-paradigm baseline comparing, statistical validation, and disclosure of reproducibility; moderate benchmarking, where multiple baselines are compared but do not offer much support of reproducibility; and weak benchmarking, where only intra-paradigm or single-baseline compare These standards create a common point of evaluation of the quality of assessments of studies based on reporting indicators that can be observed, as presented in Section 4.2.

Lastly, the framework has a reproducibility dimension that assesses the extent to which studies provide sufficient information to enable the reproduction of their results. This, as is denoted in Figure 9, encompasses code availability, configuration disclosure, workload description, and reporting of random seeds and hardware environments. The framework can be used to answer the limitations on reproducibility found in Section 4.6, where 53.4% of studies used custom simulators and 28.16% failed to identify their evaluation environment by explicitly incorporating these elements as a classifiable dimension.

6.2 Standardized Benchmarking Protocol Layer

In addition to the taxonomy layer, the proposed framework presents a structured benchmarking protocol to enable more uniform and open evaluation practices across studies. The protocol has

five suggested steps arranged in order but flexible across various experimental settings, as shown in Phase 2 of Figure 9.

The former one involves the selection and standardization of datasets. It is recommended that studies use at least one real-world or trace workload to supplement synthetic workloads where feasible, and clearly report key workload properties, including arrival patterns, heterogeneity, and system scale. The rationale for this step is that 82 % of the reviewed studies used only synthetic workloads, which is statistically associated with less rigorous benchmarking (Section 4.4).

The second step is dedicated to the choice of the baseline. In order to reduce the intra-paradigm bias the framework suggests having at least one of the three methodological categories (classical scheduling strategies e.g., FCFS, Round Robin), heuristic or metaheuristic strategies, and learning-based strategies. This minimum requirement, as shown in Figure 9, makes possible broader cross-paradigm comparisons, and overcomes the pattern shown in Section 4.3, where less than 10 per cent. of the studies conducted paradigm-balanced assessments.

The third level concerns objective reporting. Studies should report results in multi-objective dimensions and show trade-offs among them, rather than reporting individual performance measures. This may be Pareto front analysis or weighted multi-objective analysis particularly when the objectives have not been reflected in the literature well, e.g. reliability and SLA compliance (Section 4.5).

The fourth step concerns statistical validation. The protocol indicates that, to enhance the plausibility of the reported findings, repeated experimental runs with appropriate statistical summaries, including confidence intervals and, where appropriate, significant tests, must be employed.

Finally, the fifth step increases the likelihood of reproducibility by making it easier to share experimental artefacts, such as implementation code, configuration information, and random seed information, allowing other researchers to reproduce the work. This can be attributed to the fact that in Section 4.6, 53.4% of the studies utilized custom environments, and 28.16 did not specify their evaluation platform.

Overall, this protocol is expected to serve as an effective guideline for creating more similar and consistent evaluation systems, rather than a mandatory and strict procedure.

6.3 Framework Advantages

Besides the taxonomy layer, the suggested framework introduces a well-structured benchmarking protocol to facilitate more standard and open evaluation practices across studies. The protocol suggests five recommended steps that are ordered but can be adaptable under different experimental conditions as indicated in Phase 2 of Figure 9.

The first one is connected with the choice and standardization of datasets. Studies are advised to employ at least one real-world or trace workload to augment synthetic workloads where possible and to explicitly report important workload characteristics, such as arrival patterns, heterogeneity,

and system scale. The reasoning behind this move is that 82 of the studies reviewed employed synthetic workloads exclusively, which is statistically related to less rigorous benchmarking (Section 4.4).

The second step is dedicated to the choice of the baseline. To reduce intra-paradigm bias, the framework proposes to incorporate at least one baseline that belongs to three methodological groups: classical strategizing approaches (e.g., FCFS, Round Robin), heuristic or metaheuristic approaches, and learning-based approaches. As Figure 9 shows, this minimum condition makes it easier to make broader cross-paradigm comparisons and addresses the pattern shown in Section 4.3, where less than 10% of the studies conducted paradigm-balanced assessments.

The third level concerns objective reporting. Rather than reporting single performance measures, the multi-objective dimensions should be reported in the studies and trade-offs between them should be demonstrated. This can be Pareto front analysis or weighted multi-objective evaluation, especially when the objectives are underrepresented in the literature, e.g., reliability and SLA compliance (Section 4.5).

The fourth step relates to statistical validation. The protocol suggests that, to increase the credibility of the reported results, repeated experimental runs with relevant statistical summaries, such as confidence intervals and, where relevant, significant tests, should be used.

Lastly, the fifth step enhances reproducibility by facilitating the dissemination of experimental artefacts, such as implementation code, configuration information, and random seed information, enabling other researchers to reproduce the work. This is due to the simulator fragmentation mentioned in Section 4.6, in which 53.4% of the studies used custom environments, and 28.16% did not report their evaluation platform.

On the whole, this protocol is supposed to be a useful guideline in developing more similar and consistent evaluation systems, but not a rigid or compulsory process.

6.4 Illustrative Application Scenario

Figure 9 demonstrates that the UTBF application workflow incorporates taxonomy classification and benchmarking protocol steps into a single evaluation process.

To demonstrate the application of UTBF in practice, this section presents it as a real classification and evaluation analysis of an existing study based on the reviewed corpus. The chosen article is Kardani-Moghaddam et al. (2021) [115], which proposes ADRL, a hybrid anomaly-aware deep reinforcement learning model for cloud environment resource scaling. The study was chosen because it presents a methodology profile across several UTBF dimensions, making it a representative and informative application case.

The study is categorised at the taxonomy level according to the framework's five dimensions as follows. Regarding the learning paradigm, ADRL is a hybrid methodology that integrates supervised anomaly detection with deep reinforcement learning and falls under the ML+DRL category in the taxonomy. Its objective model is truly a multi-dimensional one, with its latency, energy consumption, QoS, cost, resource utilization, reliability, and SLA compliance (6 out of 10 objectives being monitored in the framework), and it includes two of the least represented objectives in the literature reviewed (reliability and SLA compliance, respectively). The evaluation environment is based on a real-world trace dataset rather than a synthetic workload, which categorises it as one of the 9.71% of studies that use trace-based evaluation. The concept of benchmarking strength is categorised as strong, supported by comparisons with four methodologically distinct baselines: DQN, a rule-based controller, a threshold-based approach, and an LSTM-based predictor. Lastly, as far as reproducibility is concerned, the study is adequately experimental enough to interpret the results, but the implementation code is not publicly accessible.

The study follows some of the steps the framework recommends at the benchmarking protocol level. The application of the real-world trace dataset meets the dataset standardization step and mitigates the external validity issues that were found in Section 4.4. The baseline set includes several paradigms, such as a learning-based approach (DQN), a classical rule-based one, and a predictive model (LSTM), which agrees with the cross-paradigm baseline recommendation. The multi-objective reporting step is manifested in the multi-dimensional objective reporting, which is both performance-oriented and reliability-oriented metrics. However, the study does not have repeated experimental runs, confidence limits, or present any implementation code, making correspondence to the statistical validation and reproducibility steps of the protocol more challenging.

This discussion shows that UTBF can be utilised in describing the strengths and weaknesses of an existing study in a coherent evaluation framework. The framework places the study in multiple dimensions at the same time, rather than producing binary judgment, following the points of consistency with the recommended practice and the points of methodological inadequacy. In this way, UTBF may be thought of as a diagnostic, but not a ranking, of its kind - it can be used to give systematic interpretation without the necessity to reduce the quality of the study to one number.

7. Threats to Validity

Various threats to validity are possible in this study's findings and conclusions and ought to be taken into account when interpreting them.

7.1 Selection Bias

The existing study is based on a filtered set of 103 articles on the topic of task scheduling using ML on edge clouds. Although systematic inclusion criteria were employed, some of the relevant studies may not have been found because of differences in database coverage, formulation of keywords or indexing of publications. This risk was addressed by using general search terms to cover DRL, ML, metaheuristics and hybrid scheduling techniques, inclusion of journal and high-quality conference papers, and similar screening and categorization criteria throughout the study selection.

7.2 Classification Bias

Since analysis is based on structured manual encoding of published studies, there may be some residual classification bias. Misclassification may happen when articles report on their approach, assessment environments, or benchmarking information so incompletely or vaguely. To minimise this risk, explicit indicators, such as the availability of statistical testing, cross-paradigm baselines, the type of datasets and the use of simulators, were used to guide the coding process and were directly born out of the reported methodology and evaluation sections. Besides, all entries were normalised in a structured taxonomy template.

7.3 Reporting Bias in Primary Studies

Another weakness is the lack of reported primary studies. Other papers are silent on simulator settings, parameter settings, workload characteristics and implementation. Consequently, some of the benchmarking rigour analysis could indicate the quality of reporting rather than the strength of the methodological aspects. Meanwhile, partial reporting is pertinent to the evaluation of reproducibility and thus a significant part of the assessment.

7.4 Temporal Bias

The dataset has a skewed temporal distribution of studies, with the number of publications rising over the last few years. This can influence how temporal trends are interpreted, particularly in early years with small sample sizes or incomplete years, e.g., 2026. To limit overinterpretation, temporal observations were approached more decisively during lean periods, and more focus was on the more stable period of publication between 2019 and 2025.

7.5 Framework Generalization Limitations

The proposed Unified Taxonomy and Benchmarking Framework (UTBF) is designed for use in edge-cloud-based ML task scheduling. Though some of its principles can be applicable in other distributed computing settings, its relevance beyond this field would have to be investigated empirically. In this regard, the framework should be construed as rather a guide to this field of research than a world-tested paradigm of all scheduling problems.

8. Conclusion

The authors presented a data-driven literature review on the topic of ML-based task scheduling in edge-cloud computing environments in this article with a selected list of 103 studies representing different methodological paradigms. Structured quantitative analysis and inferential statistical testing of the research paper revealed that several common methodological gaps recur, including dataset realism, benchmarking rigour, objective diversity, and experimental fragmentation.
The findings indicate a high dependency on synthetic workloads, an imbalanced benchmarking across paradigms, little cross-paradigm comparison and a large variance in evaluation environments. The statistical analysis also shows that these differences are not randomly related to the learning paradigms and the choice of dataset, but they are part of a systematic association of specific differences. Besides, despite being extensively reported, the objectives considered in multi-objective optimization are still focused on performance-related devices like latency and QoS, whereas reliability- and SLA-related goals are far underrepresented.

The paper, in response to these observations, proposed a Unified Taxonomy and Benchmarking Framework (UTBF), which integrates a structured classification framework with a benchmarking-oriented evaluation perspective. The framework is not to be viewed as a strictly descriptive taxonomy but is instead designed to facilitate more transparent reporting, more consistent comparisons, and enhanced reproducibility across studies.

In general, the offered framework is a systematic approach to interpreting and analysing ML-based task-scheduling studies in edge-cloud applications. Future research can further explore its applicability by extending its use in more empirical ways, such as trace-driven studies, real-world data, and more publicly accessible benchmarking materials.

Data Availability Statement

The data analysed in this paper were created by hand, as the authors used a sample of 103 peer-reviewed articles on machine learning-assisted task scheduling in edge clouds. It also includes systematic annotations at various methodological and evaluative levels, such as learning paradigms, the strength of benchmarking, the nature of the dataset, and evaluation environments.
All data processing scripts, the curated dataset, and the coding manual are maintained in a private GitHub repository during the review process and will be made publicly available upon acceptance of the manuscript.
https://github.com/nagwaelmobark/ml-edge-cloud-taxonomy

Conflict of Interest

The authors declare no conflicts of interest.

Ethics Statement

This study is based on the analysis of previously published research articles and does not involve human participants or personal data.

9. References

[1] N. Elmobark, H. El-Ghareeb, and S. S. Elhishi, BlueEdge neural network approach and its application to automated data type classification in mobile edge computing," Scientific Reports, vol. 15, no. 1, Dec. 2025, Doi: 10.1038/s41598-025-30445-z.

[2] N. Elmobark, "Intelligent Edges: Mapping the Future Convergence of Edge Computing and Big Data Analytics," Journal of Science and Technology, vol. 30, no. 3, Jan. 2025, Doi: 10.20428/jst.v30i3.

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, Oct. 2016, Doi: 10.1109/JIOT.2016.2579198.

[4] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in Proc. ACM Workshop on Hot Topics in Networks (HotNets), 2016, Doi: 10.1145/3005745.3005750.

[5] Y. Chen, N. Zhang, Y. Zhang, X. Shen, et al., "Energy Efficient Dynamic Offloading in Mobile Edge Computing for Internet of Things," IEEE Transactions on Cloud Computing, pp. 1–1, Feb. 2019, Doi: 10.1109/TCC.2019.2898657.

[6] N. Elmobark, "Evaluating the trade-offs between machine learning and deep learning: A multi-dimensional analysis," Journal of Computer Software and Program (JCSP), vol. 2, no. 1, pp. 10–18, Mar. 2025, Doi: 10.69739/jcsp.v2i1.254.

[7] H. Topcuoglu, S. Hariri, and M.-Y. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 3, pp. 260–274, Apr. 2002, Doi: 10.1109/71.993206.

[8] M. Maheswaran, S. Ali, H. J. Siegel, D. Hensgen, and R. F. Freund, "Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems," in Proc. Eighth Heterogeneous Computing Workshop (HCW'99), San Juan, PR, USA, 1999, pp. 30–44, Doi: 10.1109/HCW.1999.765094.

[9] [5] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proc. ICNN'95 - Int. Conf. Neural Networks, Perth, WA, Australia, 1995, vol. 4, pp. 1942–1948, Doi: 10.1109/ICNN.1995.488968.

[10] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.

[11] S. Moharana, A. R. Swain, and G. B. Mund, “Prediction Based Task Scheduling for Load Balancing in Cloud Environment,” in Innovations in Computer Science and Engineering, Proc. 8th ICICSE, 2021, Doi: 10.1007/978-981-33-4543-0_76.

[12] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, “Resource Management with Deep Reinforcement Learning,” in Proc. 15th ACM Workshop on Hot Topics in Networks (HotNets ’16), 2016, pp. 50–56, Doi: 10.1145/3005745.3005750.

[13] L. Huang, S. Bi, and Y.-J. A. Zhang, “Deep Reinforcement Learning for Online Computation Offloading in Wireless Powered Mobile-Edge Computing Networks,” IEEE Transactions on Mobile Computing, vol. 19, no. 11, pp. 2581–2593, Nov. 2020, Doi: 10.1109/TMC.2019.2928811.

[14] W. Yu et al., “A Survey on the Edge Computing for the Internet of Things,” IEEE Access, vol. 6, pp. 6900–6919, 2018, Doi: 10.1109/ACCESS.2017.2778504.

[15] P. Mach and Z. Becvar, “Mobile Edge Computing: A Survey on Architecture and Computation Offloading,” IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017, Doi: 10.1109/COMST.2017.2682318.

[16] A. A. Ismail, N. E. Khalifa, and R. A. El-Khoribi, “A survey on resource scheduling approaches in multi-access edge computing environment: A deep reinforcement learning study,” *Cluster Computing*, vol. 28, p. 184, 2025, Doi: 10.1007/s10586-024-04893-7.

[17] M. A. Islam, S. Razzaque, M. Hassan, W. N. Ismail, and B. Song, “A survey on scheduling techniques in the edge cloud, issues, challenges and future directions,” *arXiv preprint* arXiv:2202.07799v2, Feb. 2022. [Online]. Available: https://arxiv.org/abs/2202.07799

[18] Y. Zhang, X. Chen, and L. Wang, “Edge-cloud collaborative computing on distributed intelligence and model optimization: A survey,” *arXiv preprint* arXiv:2505.01821v1, May 2025. [Online]. Available: https://arxiv.org/abs/2505.01821

[19] S. Dong, J. Tang, K. Abbas, R. Hou, J. Kamruzzaman, L. Rutkowski, and R. Buyya, “Task offloading strategies for mobile edge computing: A survey,” *Computer Networks*, vol. 246, p. 110791, 2024, Doi: 10.1016/j.comnet.2024.110791.

[20] P. Singh, R. Kumar, and S. Gupta, “A survey on task scheduling in edge-cloud,” *SN Computer Science*, vol. 6, no. 2, pp. 1–18, 2025, Doi: 10.1007/s42979-025-03757-0.

[21] S. Chen, Q. Li, M. Zhou, and A. Abusorrah, “Recent advances in collaborative scheduling of computing tasks in an edge computing paradigm,” *Sensors*, vol. 21, no. 3, p. 779, Jan. 2021, Doi: 10.3390/s21030779.

[22] D. Bodra and S. Khairnar, “Machine learning-based cloud resource allocation algorithms: A comprehensive comparative review,” *Frontiers in Computer Science*, vol. 7, Oct. 2025, Doi: 10.3389/fcomp.2025.1678976.

[23] X. Chen *et al*., “Deep reinforcement learning for job scheduling and resource management in cloud computing: An algorithm-level review,” *arXiv preprint* arXiv:2501.01007, Jan. 2025. [Online]. Available: https://arxiv.org/abs/2501.01007

[24] E. H. Houssein, A. G. Gad, Y. M. Wazery, and P. N. Suganthan, “Task scheduling in cloud computing based on meta-heuristics: Review, taxonomy, open challenges, and future trends,” *Swarm and Evolutionary Computation*, vol. 62, p. 100841, 2021, Doi: 10.1016/j.swevo.2021.100841.

[25] Y. Qawqzeh, M. T. Alharbi, A. Jaradat, and K. N. Abdul Sattar, “A review of swarm intelligence algorithms deployment for scheduling and optimization in cloud computing environments,” *PeerJ Computer Science*, vol. 7, e696, 2021, Doi: 10.7717/peerj-cs.696.

[26] J. Yoon, Y.-H. Kim, and C.-C. Lin, “A novel approach to the job shop scheduling problem based on the cooperative scheduling system and deep Q-network in multi-access edge computing,” *Sensors*, vol. 21, no. 13, p. 4553, 2021, Doi: 10.3390/s21134553.

[27] C.-C. Lin, D.-J. Deng, Y.-L. Chih, and H.-T. Chiu, “Smart manufacturing scheduling with edge computing using multiclass deep Q network,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4276–4284, Jul. 2019, Doi: 10.1109/TII.2019.2908210.

[28] L. Zeng, Q. Liu, S. Shen, and X. Liu, “Improved double deep Q network-based task scheduling algorithm in edge computing for makespan optimization,” *Tsinghua Science and Technology*, vol. 29, no. 3, pp. 806–817, Jun. 2024, Doi: 10.26599/TST.2023.9010058.

[29] T. Du, C. Li, and Y. Luo, “Latency-aware computation offloading and DQN-based resource allocation approaches in SDN-enabled MEC,” *Ad Hoc Networks*, 2022, Doi: 10.1016/j.adhoc.2022.102950.

[30] X. Hu and Y. Huang, “Deep reinforcement learning based offloading decision algorithm for vehicular edge computing,” *PeerJ Computer Science*, vol. 8, e1126, Oct. 2022, Doi: 10.7717/peerj-cs.1126.

[31] Y. He, C. Liang, F. R. Yu, N. Zhao, and H. Yin, “Deep-reinforcement-learning-based offloading scheduling for vehicular edge computing,” IEEE Internet of Things Journal, vol. 7, no. 6, pp. 5449–5465, Jun. 2020, Doi: 10.1109/JIOT.2020.2978830.

[32] W. Shi, L. Chen, and X. Zhu, “Task offloading decision-making algorithm for vehicular edge computing: A deep-reinforcement-learning-based approach,” *Sensors*, vol. 23, no. 17, p. 7595, Aug. 2023, Doi: 10.3390/s23177595.

[33] G. Nieto, I. de la Iglesia, U. Lopez-Novoa *et al*., “Deep reinforcement learning techniques for dynamic task offloading in the 5G edge-cloud continuum,” *Journal of Cloud Computing*, vol. 13, p. 94, 2024, Doi: 10.1186/s13677-024-00658-0.

[34] M. Deng, W. Liang, B. Qiu, and X. Xie, “Federated deep reinforcement learning-based task offloading strategy with game theory in vehicular edge computing,” *Physical Communication*, vol. 72, p. 102802, 2025, Doi: 10.1016/j.phycom.2025.102802.

[35] M. Xie, J. Ye, G. Zhang, and X. Ni, “Deep reinforcement learning-based computation offloading and distributed edge service caching for mobile edge computing,” *Computer Networks*, vol. 250, p. 110564, 2024, Doi: 10.1016/j.comnet.2024.110564.

[36] Y. Fan and X. Cai, “A deep reinforcement approach for computation offloading in MEC dynamic networks,” *EURASIP Journal on Advances in Signal Processing*, vol. 2024, p. 48, 2024, Doi: 10.1186/s13634-024-01142-2.

[37] L. Ale, S. A. King, N. Zhang, A. R. Sattar, and J. Skandaraniyam, “D3PG: Dirichlet DDPG for task partitioning and offloading with constrained hybrid action space in mobile-edge computing,” *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 19260–19272, Oct. 2022, Doi: 10.1109/JIOT.2022.3166110.

[38] X. Chen and G. Liu, “Federated deep reinforcement learning-based task offloading and resource allocation for smart cities in a mobile edge network,” *Sensors*, vol. 22, no. 13, p. 4738, Jun. 2022, Doi: 10.3390/s22134738.

[39] X. Zhao, Y. Wu, T. Zhao, F. Wang, and M. Li, “Federated deep reinforcement learning for task offloading and resource allocation in mobile edge computing-assisted vehicular networks,” *Journal of Network and Computer Applications*, vol. 229, p. 103941, 2024, Doi: 10.1016/j.jnca.2024.103941.

[40] L. Lin, W. Zhou, Z. Yang, and J. Liu, “Deep reinforcement learning-based task scheduling and resource allocation for NOMA-MEC in Industrial Internet of Things,” *Peer-to-Peer Networking and Applications*, 2022, Doi: 10.1007/s12083-022-01348-x.

[41] Y. Fan, J. Ge, S. Zhang, J. Wu, and B. Luo, “Decentralized scheduling for concurrent tasks in mobile edge computing via deep reinforcement learning,” *IEEE Transactions on Mobile Computing*, vol. 23, no. 4, pp. 2765–2779, Apr. 2024.

[42] L.-T. Hsieh, H. Liu, Y. Guo, and R. Gazda, “Deep reinforcement learning-based task assignment for cooperative mobile edge computing,” *IEEE Transactions on Mobile Computing*, vol. 23, no. 4, pp. 3156–3171, Apr. 2024, Doi: 10.1109/TMC.2023.3270242.

[43] D. Wei, J. Wen, P. Xu, H. Shi, and C. Pan, “Dependent tasks joint scheduling and offloading for edge computing based on deep reinforcement learning,” *Ad Hoc Networks*, vol. 183, p. 104111, 2026, Doi: 10.1016/j.adhoc.2025.104111.

[44] N. Wang, S. Pang, X. Ji, H. Gui, and X. He, “An efficient scheduling scheme for intelligent driving tasks in a novel vehicle-edge architecture considering mobility and load balancing,” *Future Generation Computer Systems*, vol. 160, pp. 180–194, Nov. 2024, Doi: 10.1016/j.future.2024.06.020.

[45] M. R. Raju, S. K. Mothku, and M. K. Somesula, “DRL-based task scheduling scheme in vehicular fog computing: Cooperative and mobility aware approach,” *Ad Hoc Networks*, vol. 173, p. 103819, 2025, Doi: 10.1016/j.adhoc.2025.103819.

[46] Q. Tang *et al*., “Distributed task scheduling in serverless edge computing networks for the Internet of Things: A learning approach,” *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 19634–19648, Oct. 2022, Doi: 10.1109/JIOT.2022.3167417.

[47] S. Swarup, E. M. Shakshuki, and A. Yasar, “Energy efficient task scheduling in fog environment using deep reinforcement learning approach,” *Procedia Computer Science*, vol. 191, pp. 65–75, 2021, Doi: 10.1016/j.procs.2021.07.012.

[48] Y. Sun, W. Yu, X. Zhou, J. Zhao, and X. Zhang, “Multi-Agent Deep Reinforcement Learning for Device-enhanced Distributed Task Scheduling in Terminal-edge Collaborative Computing Networks,” *IEEE Internet of Things Journal*, published online Dec. 22, 2025, Doi: 10.1109/JIOT.2025.3646683.

[49] S. Tan, T. Huang, E. Zhu, J. Qin, and X. Fan, “EPRS: Experience-prioritized reinforcement scheduler in edge clusters,” *Sensors*, vol. 26, no. 4, p. 1168, 2026, Doi: 10.3390/s26041168.

[50] M. Tang and V. W. S. Wong, “Deep reinforcement learning for task offloading in mobile edge computing systems,” *IEEE Transactions on Mobile Computing*, vol. 21, no. 6, pp. 1985–1997, Jun. 2022, Doi: 10.1109/TMC.2020.3036871.

[51] L. Liu, J. Feng, X. Mu, Q. Pei, D. Lan, and M. Xiao, “Asynchronous deep reinforcement learning for collaborative task computing and on-demand resource allocation in vehicular edge computing,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 15513–15526, Dec. 2023, Doi: 10.1109/TITS.2023.3249745.

[52] P. Zhang, S. Li, L. Tan, N. Kumar, J. Wang, and K. Liu, “In-situ data scheduling optimization based on rainbow DQN for IIoT,” *Future Generation Computer Systems*, vol. 174, p. 107940, 2026, Doi: 10.1016/j.future.2025.107940.

[53] N. Yamsani and P. C. Reddy, “EdgeSched-DQN: An intelligent deep reinforcement learning-based framework for optimized task scheduling in edge-cloud environments,” *Array*, vol. 29, p. 100645, 2026, Doi: 10.1016/j.array.2025.100645.

[54] S. Sheng, P. Chen, Z. Chen, L. Wu, and Y. Yao, “Deep reinforcement learning-based task scheduling in IoT edge computing,” *Sensors*, vol. 21, no. 5, p. 1666, Feb. 2021, Doi: 10.3390/s21051666.

[55] L. Zhai, Z. Lu, J. Sun, and X. Li, “Joint task offloading and computing resource allocation with DQN for task-dependency in multi-access edge computing,” *Computer Networks*, vol. 263, p. 111222, 2025, Doi: 10.1016/j.comnet.2025.111222.

[56] V. Mnih, A. Puigdomènech Badia, M. Mirza, K. Kavukcuoglu, *et al*., “Asynchronous methods for deep reinforcement learning,” *arXiv preprint* arXiv:1602.01783, Feb. 2016, Doi: 10.48550/arXiv.1602.01783.

[57] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint* arXiv:1707.06347, Aug. 2017, Doi: 10.48550/arXiv.1707.06347.

[58] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” *arXiv preprint* arXiv:1801.01290, 2018, Doi: 10.48550/arXiv.1801.01290.

[59] S. Fujimoto, H. van Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” *arXiv preprint* arXiv:1802.09477, 2018, Doi: 10.48550/arXiv.1802.09477.

[60] T. P. Lillicrap *et al*., “Continuous control with deep reinforcement learning,” *arXiv preprint* arXiv:1509.02971, 2015, Doi: 10.48550/arXiv.1509.02971.

[61] A. Qadeer and M. J. Lee, “DDPG-Edge-Cloud: A deep-deterministic policy gradient based multi-resource allocation in edge-cloud system,” in *Proc. IEEE International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Jeju Island, South Korea, Feb. 2022, pp. 339–344, Doi: 10.1109/ICAIIC54071.2022.9722676.

[62] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, “Multi-agent actor-critic for mixed cooperative-competitive environments,” in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 6382–6393, Doi: 10.48550/arXiv.1706.02275.

[63] S. Tuli, S. Ilager, K. Ramamohanarao, and R. Buyya, “Dynamic scheduling for stochastic edge-cloud computing environments using A3C learning and residual recurrent neural networks,” *IEEE Transactions on Mobile Computing*, vol. 21, no. 3, pp. 940–954, Mar. 2022, Doi: 10.1109/TMC.2020.3017079.

[64] Y. Zou, J. Rao, and T. He, “A3C-DO: A regional resource scheduling framework based on deep reinforcement learning in edge scenario,” *IEEE Transactions on Computers*, vol. 70, no. 2, pp. 228–239, Feb. 2021, Doi: 10.1109/TC.2020.2987567.

[65] Y. Yu, Y. Yan, S. Li, Z. Li, and D. Wu, “Task offloading scheme based on proximal policy optimization algorithm,” *Applied Sciences*, vol. 15, no. 9, p. 4761, Apr. 2025, Doi: 10.3390/app15094761.

[66] Y. He, C. Liang, F. R. Yu, N. Zhao, and H. Yin, “Deep-reinforcement-learning-based offloading scheduling for vehicular edge computing,” *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5449–5465, Jun. 2020, Doi: 10.1109/JIOT.2020.2978830.

[67] C. Zhang, C. Wu, M. Lin, Y. Lin, and W. Liu, “Proximal policy optimization for efficient D2D-assisted computation offloading and resource allocation in multi-access edge computing,” *Future Internet*, vol. 16, no. 1, p. 19, Jan. 2024, Doi: 10.3390/fi16010019.

[68] C. Sun, X. Wu, X. Li, Q. Fan, J. Wen, and V. C. M. Leung, “Cooperative computation offloading for multi-access edge computing in 6G mobile networks via soft actor critic,” *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 6, pp. 5601–5614, 2024, Doi: 10.1109/TNSE.2021.3076795.

[69] Z. Chen, H. Hu, J. Min, G. Luo, and T. El-Ghazawi, “Adaptive and efficient resource allocation in cloud datacenters using actor-critic deep reinforcement learning,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 8, pp. 1911–1923, Aug. 2022, Doi: 10.1109/TPDS.2021.3132422.

[70] Q. Fang, Z. Xing, Q. Liao, S. Shi, and B. Ye, “An improved soft actor-critic task offloading and edge computing resource allocation algorithm for image segmentation tasks in the internet of vehicles,” *World Electric Vehicle Journal*, vol. 16, no. 7, p. 353, Jun. 2025, Doi: 10.3390/wevj16070353.

[71] Q. Tang, R. Xie, Z. Fang, T. Huang, T. Chen, R. Zhang, and F.-R. Yu, “Joint service deployment and task scheduling for satellite edge computing: A two-timescale hierarchical approach,” *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 5, pp. 1063–1079, May 2024, Doi: 10.1109/JSAC.2024.3365889.

[72] A. M. Seid *et al*., “Multi-Agent DRL for task offloading and resource allocation in multi-UAV enabled IoT edge network,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4651–4665, Dec. 2021, Doi: 10.1109/TNSM.2021.3096673.

[73] A. Suzuki, M. Kobayashi, and E. Oki, “Multi-agent deep reinforcement learning for cooperative computing offloading and route optimization in multi cloud-edge networks,” *IEEE Transactions on Network and Service Management*, vol. 20, no. 4, pp. 4416–4434, Dec. 2023, Doi: 10.1109/TNSM.2023.3267809.

[74] X. Chen and D. Long, “Task scheduling of cloud computing using integrated particle swarm algorithm and ant colony algorithm,” *Cluster Computing*, vol. 22, no. 4, pp. 2761–2769, 2017, Doi: 10.1007/s10586-017-1479-y.

[75] S. D. Vispute and P. Vashisht, “Energy-efficient task scheduling in fog computing based on particle swarm optimization,” *SN Computer Science*, vol. 4, p. 391, 2023, Doi: 10.1007/s42979-022-01639-3.

[76] A. Singh, R. Kumar, and S. Gupta, “An ant colony optimization based job allocation strategy for IoT-edge-cloud computing environment,” *Journal of Ambient Intelligence and Humanized Computing*, 2025, Doi: 10.1007/s12652-025-05023-w.

[77] Y. Zhang, L. Wang, and X. Chen, “Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, pp. 3859–3875, 2020, Doi: 10.1007/s12652-020-02614-7.

[78] X. Zhang, “A fine-grained task scheduling mechanism for digital economy services based on intelligent edge and cloud computing,” *Journal of Cloud Computing*, vol. 12, p. 30, 2023, Doi: 10.1186/s13677-023-00402-0.

[79] M. Gamal, R. Rizk, H. Mahdi, and B. Elhady, “Task scheduling of cloud computing based on hybrid particle swarm algorithm and genetic algorithm,” *Cluster Computing*, vol. 24, pp. 2245–2259, 2020, Doi: 10.1007/s10586-020-03221-z.

[80] A. Mahjoubi, K.-J. Grinnemo, and J. Taheri, “EHGA: A genetic algorithm based approach for scheduling tasks on distributed edge-cloud infrastructures,” in *Proc. 2022 IEEE 13th International Conference on Network of the Future (NoF)*, Ghent, Belgium, Oct. 2022, pp. 1–5.

[81] A. Mahjoubi, K.-J. Grinnemo, and J. Taheri, “An efficient simulated annealing-based task scheduling technique for task offloading in a mobile edge architecture,” in *Proc. 2022 IEEE 11th International Conference on Cloud Networking (CloudNet)*, Paris, France, Nov. 2022, pp. 159–167, Doi: 10.1109/CloudNet55617.2022.9978900.

[82] N. Bacanin, M. Zivkovic, T. Bezdan, K. Venkatachalam, and M. Abouhawwash, “Modified firefly algorithm for workflow scheduling in cloud-edge environment,” *Neural Computing and Applications*, vol. 34, pp. 9043–9068, 2022, Doi: 10.1007/s00521-022-06925-y.

[83] S. Mangalampalli, G. R. Karri, and A. A. Elngar, “An efficient trust-aware task scheduling algorithm in cloud computing using firefly optimization,” *Sensors*, vol. 23, no. 3, p. 1384, Jan. 2023, Doi: 10.3390/s23031384.

[84] I. Nujhat *et al*., “Task offloading exploiting grey wolf optimization in collaborative edge computing,” *Journal of Cloud Computing*, vol. 13, p. 23, 2024, Doi: 10.1186/s13677-023-00570-z.

[85] A. Asghari *et al*., “Metaheuristic-based task scheduling for latency-sensitive IoT applications in edge computing,” *Cluster Computing*, 2024, Doi: 10.1007/s10586-024-04878-6.

[86] F. A. Saif, R. Latip, Z. M. Hanapi, and K. Shafinah, “Modified grey wolf optimization for energy-efficient internet of things task scheduling in fog computing,” *Scientific Reports*, 2025, Doi: 10.1038/s41598-025-99837-5.

[87] C. Chandrashekar *et al*., “An enhanced whale optimization algorithm for task scheduling in edge computing environments,” *Frontiers in Big Data*, vol. 7, 2024, Doi: 10.3389/fdata.2024.1422546.

[88] M. K. Jha and M. Kumar, “An autonomic offloading and resource allocation technique for IoT applications in edge computing,” *Journal of Supercomputing*, 2025, Doi: 10.1007/s11227-024-06491-1.

[89] M. Bakshi, C. Chowdhury, and U. Maulik, “Cuckoo search optimization-based energy efficient job scheduling approach for IoT-edge environment,” *Journal of Supercomputing*, 2023, Doi: 10.1007/s11227-023-05358-1.

[90] A. Yousif, M. B. Bashir, and A. Ali, “An evolutionary algorithm for task clustering and scheduling in IoT edge computing,” *Mathematics*, vol. 12, no. 2, p. 281, 2024, Doi: 10.3390/math12020281.

[91] X. Wang, Y. Chen, and L. Zhang, “Combined particle swarm optimization and ant colony system for energy efficient cloud data centers,” *Concurrency and Computation: Practice and Experience*, vol. 33, no. 12, p. e6195, 2021, Doi: 10.1002/cpe.6195.

[92] S. Patel, M. Singh, and R. Kumar, “Efficient task scheduling on the cloud using artificial neural network and particle swarm optimization,” *Concurrency and Computation: Practice and Experience*, vol. 36, no. 8, p. e7954, 2024, Doi: 10.1002/cpe.7954.

[93] D. Saxena and A. K. Singh, “Performance analysis of machine learning centered workload prediction models for cloud,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 4, pp. 1207–1224, Apr. 2023, Doi: 10.1109/TPDS.2023.3240567.

[94] J. Bi, H. Ma, H. Yuan, and J. Zhang, “Accurate prediction of workloads and resources with multi-head attention and hybrid LSTM for cloud data centers,” *IEEE Transactions on Sustainable Computing*, vol. 8, no. 3, pp. 375–384, Jul.–Sep. 2023, Doi: 10.1109/TSUSC.2023.3259522.

[95] J. Gao, H. Wang, and H. Shen, “Machine learning based workload prediction in cloud computing,” in *Proc. 2020 29th International Conference on Computer Communications and Networks (ICCCN)*, Honolulu, HI, USA, Aug. 2020, pp. 1–9, Doi: 10.1109/ICCCN49398.2020.9209730.

[96] Y. Tu, H. Chen, L. Yan, and X. Zhou, “Task offloading based on LSTM prediction and deep reinforcement learning for efficient edge computing in IoT,” *Future Internet*, vol. 14, no. 2, p. 30, Jan. 2022, Doi: 10.3390/fi14020030.

[97] M. Chen and Y. Hao, “Intelligent task prediction and computation offloading based on mobile-edge cloud computing,” *Future Generation Computer Systems*, vol. 102, pp. 925–931, Jan. 2020, Doi: 10.1016/j.future.2019.09.035.

[98] G. Vijayasekaran and M. Duraipandian, “An efficient clustering and deep learning based resource scheduling for edge computing to integrate cloud-IoT,” *Wireless Personal Communications*, vol. 124, no. 3, pp. 2029–2044, 2022, Doi: 10.1007/s11277-021-09442-8.

[99] M. Salimian, H. Karimian, M. R. Salimian, and A. Rezaei, “Technical study of deep learning in cloud computing for accurate workload prediction,” *Electronics*, vol. 12, no. 3, p. 650, Jan. 2023, Doi: 10.3390/electronics12030650.

[100] I. Ullah and H. Y. Youn, “Task classification and scheduling based on K-means clustering for edge computing,” *Wireless Personal Communications*, vol. 113, no. 4, pp. 2611–2624, 2020, Doi: 10.1007/s11277-020-07343-w.

[101] M. S. Sheikh, R. N. Enam, and R. I. Qureshi, “Machine learning-driven task scheduling with dynamic K-means based clustering algorithm using fuzzy logic in fog environment,” *Frontiers in Computer Science*, vol. 5, p. 1293209, Dec. 2023, Doi: 10.3389/fcomp.2023.1293209.

[102] J. Ma *et al*., “Task offloading optimization in mobile edge computing based on a deep reinforcement learning algorithm using density clustering and ensemble learning,” *Scientific Reports*, vol. 15, p. 184, Jan. 2025, Doi: 10.1038/s41598-024-84038-3.

[103] T. Khan, W. Tian, G. Zhou, S. Ilager, M. Gong, and R. Buyya, “Machine learning (ML)-centric resource management in cloud computing: A review and future directions,” *Journal of Network and Computer Applications*, vol. 204, p. 103405, Jul. 2022, Doi: 10.1016/j.jnca.2022.103405.

[104] S. C. Sonmez, A. Ozgovde, and C. Ersoy, “Machine learning-based workload orchestrator for vehicular edge computing,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2239–2251, Apr. 2021, Doi: 10.1109/TITS.2020.3024233.

[105] B. Wang, Z. Zhang, Z. Song, Y. Chen, C. Zhou, and Y. Chu, “A hybrid PSO and GA algorithm with rescheduling for task offloading in device-edge-cloud collaborative computing,” *Cluster Computing*, 2025, Doi: 10.1007/s10586-024-04851-3.

[106] Z. Liao, X. Zhao, Y. Zhang, and J. Wang, “A mobility-aware task scheduling by hybrid PSO and GA for mobile edge computing,” *Cluster Computing*, 2024, doi: 10.1007/s10586-024-04341-6.

[107] H. Yuan, Z. Zheng, J. Bi, J. Zhang, and M. C. Zhou, “Energy-optimized task offloading with genetic simulated-annealing-based PSO for heterogeneous edge and cloud computing,” in *Proc. IEEE SMC*, Kuching, Malaysia, Oct. 2024, pp. 647–652, Doi: 10.1109/SMC54092.2024.10831521.

[108] X. Li, J. Li, X. Duan, and J. Xia, “Hybrid deep reinforcement learning and genetic algorithm-based resource allocation framework for edge computing,” *Peer-to-Peer Networking and Applications*, 2025, Doi: 10.1007/s12083-025-02149-8.

[109] A. Xu, Z. Hu, X. Li, R. Tian, X. Zhang, B. Chen, H. Xiao, H. Zheng, X. Feng, M. Zheng, P. Zhong, and K. Li, “TransEdge: Task offloading with GNN and DRL in edge-computing-enabled transportation systems,” *IEEE Internet of Things Journal*, Dec. 2024, Doi: 10.1109/JIOT.2024.3443866.

[110] S. Shen *et al*., “EdgeMatrix: A resource-redefined scheduling framework for SLA-guaranteed multi-tier edge-cloud computing systems,” *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 3, pp. 820–834, Mar. 2023, Doi: 10.1109/JSAC.2022.3229444.

[111] A. Heidari, N. J. Navimipour, and M. Unal, “Federated reinforcement learning-based dynamic resource allocation and task scheduling in edge for IoT applications,” *Sensors*, vol. 25, no. 7, p. 2197, 2025, Doi: 10.3390/s25072197.

[112] T. M. Ho, K.-K. Nguyen, and M. Cheriet, “Federated deep reinforcement learning for task scheduling in heterogeneous autonomous robotic system,” *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 1, pp. 528–540, Jan. 2024, Doi: 10.1109/GLOBECOM48099.2022.10000980.

[113] X. Su, S. Xu, K. Li, and T. Wang, “Prediction-based resource deployment and task scheduling in edge-cloud collaborative computing,” *Wireless Communications and Mobile Computing*, vol. 2022, p. 2568503, 2022, Doi: 10.1155/2022/2568503.

[114] T. Zheng, J. Wan, J. Zhang, and Y. Jiang, “Deep reinforcement learning-based workload scheduling for edge computing,” *Journal of Cloud Computing*, vol. 11, p. 3, 2022, Doi: 10.1186/s13677-021-00276-0.

[115] S. Kardani-Moghaddam, R. Buyya, and K. Ramamohanarao, “ADRL: A hybrid anomaly-aware deep reinforcement learning-based resource scaling in clouds,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 3, pp. 514–526, Mar. 2021, Doi: 10.1109/TPDS.2020.3025914.

[116] M. S. Rama Krishna *et al*., “Reinforcement learning based multi-objective task scheduling for energy efficient and cost effective cloud edge computing,” *Scientific Reports*, vol. 15, 2025, Doi: 10.1038/s41598-025-25666-1.

[117] M. S. V. Krishna *et al*., “SLA-aware deep reinforcement learning for adaptive edge-cloud task scheduling,” *Scientific Reports*, 2025, doi: 10.1038/s41598-026-40237-8.

[118] Y. Gu, F. Liu, J. Huang, and Z. Wang, “Cost-aware cloud workflow scheduling using DRL and simulated annealing,” *Digital Communications and Networks*, 2024, Doi: 10.1016/j.dcan.2023.12.009.

[119] Z. Wang, B. Zhang, S. Yi, Y. Chen, M. Chu, and Y. Wang, “A hybrid meta-heuristic algorithm for joint edge caching and task scheduling with mobility awareness,” *Cluster Computing*, 2025, Doi: 10.1007/s10586-025-05838-4.

[120] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, “CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, Jan. 2011, Doi: 10.1002/spe.995.

[121] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, “iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments,” *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, Sep. 2017, Doi: 10.1002/spe.2509.

[122] C. Sonmez, A. Ozgovde, and C. Ersoy, “EdgeCloudSim: An environment for performance evaluation of edge computing systems,” *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 11, p. e3493, Nov. 2018, Doi: 10.1002/ett.3493.

[123] M. S. Aslanpour, S. S. Gill, and A. N. Toosi, “Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research,” *Internet of Things*, vol. 12, p. 100273, Aug. 2020, Doi: 10.1016/j.iot.2020.100273.

[124] C. Mechalikh, H. Taktak, and F. Moussa, “PureEdgeSim: A simulation framework for performance evaluation of cloud, edge and mist computing environments,” *Computer Science and Information Systems*, vol. 18, no. 1, pp. 43–66, Jan. 2021, Doi: 10.2298/CSIS200301042M.

[125] M. C. Silva Filho, R. L. Oliveira, C. C. Monteiro, P. R. M. Inácio, and M. M. Freire, “CloudSim Plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness,” in *Proc. IFIP/IEEE International Symposium on Integrated Network and Service Management (IM)*, Lisbon, Portugal, May 2017, pp. 400–406, Doi: 10.23919/INM.2017.7987304.

[126] R. Mahmud, S. Pallewatta, M. Goudarzi, and R. Buyya, “iFogSim2: An extended iFogSim simulator for mobility, clustering, and microservice management in edge and fog computing environments,” *Journal of Systems and Software*, vol. 190, p. 111351, Aug. 2022, Doi: 10.1016/j.jss.2022.111351.

[127] K. Alwasel *et al*., “IoTSim-Osmosis: A framework for modeling and simulating IoT applications over an edge-cloud continuum,” *Journal of Systems Architecture*, vol. 116, p. 101956, Jun. 2021, Doi: 10.1016/j.sysarc.2020.101956.

[128] M. Tirmazi *et al*., “Borg: The next generation,” in *Proc. EuroSys*, Heraklion, Greece, Apr. 2020, pp. 1–14, Doi: 10.1145/3342195.3387517.

[129] L. Cheng *et al*., “Characterizing and synthesizing task dependencies of data-parallel jobs in Alibaba Cloud,” in *Proc. ACM Symposium on Cloud Computing (SoCC)*, Santa Cruz, CA, USA, Nov. 2019, pp. 139–152, Doi: 10.1145/3357223.3362710.

[130] A. Qayyum *et al*., “Simulation tools for fog computing: A comparative analysis,” *Sensors*, vol. 23, no. 7, p. 3492, Mar. 2023, Doi: 10.3390/s23073492.