

A review of intelligent task scheduling models for fog computing in large-scale internet of things networks

Sababathy Palanisamy¹, Erum Ashraf², Selvakumar Manickam¹, Huda Mohammed Alsayednoor³,
Mahmood A. Al-Shareeda^{4,5}, Shankar Karuppayah¹

¹Cybersecurity Research Centre (CYRES), Universiti Sains Malaysia, George Town, Malaysia

²Faculty of Engineering Sciences and Technology, Bahria University, Islamabad, Pakistan

³Shatt Al-Arab University College, Basra, Iraq

⁴Department of Computer Engineering Technologies and Artificial Intelligence, Basrah Engineering Polytechnic College, Southern Technical University, Basra, Iraq

⁵College of Engineering, Al-Ayen University, Thi-Qar, Iraq

Article Info

Article history:

Received Apr 20, 2026

Revised Jun 23, 2026

Accepted Jul 18, 2026

Keywords:

Fog computing

Internet of things

Machine learning

Reinforcement learning

Resource management

Task scheduling

ABSTRACT

The rapid growth of internet of things (IoT) applications has generated an urgent demand for efficient task scheduling mechanisms that feature low latency, scalability, and resource awareness. While fog computing can meet this demand by extending computing and storage resources to the vicinity of data sources, it still confronts four core challenges: resource heterogeneity, dynamic workloads, mobility, and security constraints. To address these issues, this study conducts a systematic review of intelligent task scheduling for fog-enabled IoT systems. It classifies the included eligible studies into five scheduling paradigms, and innovatively proposes a cross-domain framework that integrates three types of awareness, including priority awareness. The subsequent work of this study will sort out the core strengths and limitations of existing relevant methods. Finally, open challenges and emerging research directions, including edge artificial intelligence (AI), federated learning, blockchain-assisted scheduling, and hardware-accelerated implementations are discussed. The findings provide useful guidance for developing scalable, secure, and adaptive scheduling frameworks for next-generation fog-enabled IoT systems.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Selvakumar Manickam

Cybersecurity Research Centre (CYRES), Universiti Sains Malaysia

George Town, Pulau Pinang, 11800, Malaysia

Email: Selva@usm.my

1. INTRODUCTION

The rise of the internet of things (IoT) has seen aggressive deployments of emerging infrastructure comprised by billions of devices interconnected that generate huge volumes data. In many IoT applications (e.g. smart transportation, industrial automation, healthcare monitoring, and intelligent surveillance), low latency, high reliability, and quality-of-service (QoS) guarantees are demanded [1]–[3]. The traditional cloud-centric computing architectures cannot satisfy such requirements because processing is centralized, network access can be congested, and there are long communication delays between devices and cloud data centers [4], [5].

Fog computing has recently attracted attention as a promising paradigm to overcome these limitations, and to extend computation, storage, and networking resources near data sources. Fog computing supports decentralized processing at the network edge and intermediate fog nodes, thus reducing latency, relieving

backhaul traffic, and improving context awareness [6], [7]. However, fog environments are heterogeneous in nature and thus difficult to utilize resources efficiently. One of the key aspects is task scheduling, schedulers have to adapt workloads, diverse device feature sets, unpredictable network environment, and distinct application priorities [8].

Based on the literature, at least a series of intelligent task scheduling models were proposed as it seems complex. All the mentioned above mentioned static approaches are improved in performing better scheduling of the jobs using heuristic rules, metaheuristic optimization techniques, machine learning (ML), reinforcement learning (RL), or hybrids of them all [9], [10]. While existing works show a considerable performance improvement for dedicated cases, the diversity of scheduling algorithms, optimization criteria and evaluation scenarios presents challenges in developing an overall understanding on their strengths and weaknesses as well as applicability to large-scale IoT fog systems.

Although several survey articles have reviewed task scheduling and resource management in fog and edge computing environments, most existing studies focus primarily on algorithmic categorization or specific optimization objectives such as latency, energy efficiency, or resource utilization. More recently, Chuan *et al.* [11] reviewed the challenges and opportunities associated with task scheduling in fog computing environments. Jamil *et al.* [12] presented a comprehensive taxonomy of resource allocation and task scheduling techniques in fog computing and internet of everything (IoE) environments. Hosseinzadeh *et al.* [13] conducted a systematic survey of task scheduling mechanisms in fog computing, categorizing existing solutions according to scheduling objectives, optimization techniques, and system architectures. Table 1 compares the proposed survey with representative recent review articles.

Table 1. Comparison of the proposed survey with existing review studies on fog computing task scheduling

Survey	Heuristic	ML	RL	Hybrid	Security	Locality	Priority	Future directions
Chuan <i>et al.</i> [11]	✓	✓	✓	✓	Partial	×	×	✓
Jamil <i>et al.</i> [12]	✓	✓	×	×	×	×	×	✓
Hosseinzadeh <i>et al.</i> [13]	✓	✓	✓	×	Partial	×	×	✓
This survey	✓	✓	✓	✓	✓	✓	✓	✓

Existing review papers focused on fog and edge scheduling currently suffer from two types of limitations: first, they exhibit a superficial positioning bias, and second, they have core capability shortcomings. With the emergence of new trends in related fields, such as learning-based and hybrid schedulers, it has become highly necessary to develop a new review that meets current development needs.

This study conducts a systematic review of intelligent task scheduling models for large-scale IoT networks in fog computing environments. We classify and organize existing methods, dissect their core designs, and complete a comprehensive comparison of all reviewed models. We further discuss cross-cutting intelligence and open challenges in the field, and outline clear directions for the future development of scalable, secure, and practical scheduling frameworks.

The rest of this paper is structured as follows. Section 2 provides research method in this paper. Fog computing and scheduling basics are summarized in section 3. Taxonomy of intelligent task scheduling models is discussed in section 4. We now present priority-aware, security-aware, and locality-aware scheduling concerns in the following sections. Section 5 shows priority-aware scheduling in fog computing. That is included in section 6, followed by open challenges and future research directions in section 7. Finally, section 8 concludes the paper.

2. RESEARCH METHOD

To ensure transparency, reproducibility, and scientific rigor, this review adopts the structured literature review method and sorts out relevant research on large-scale IoT intelligent task scheduling in fog computing environments in strict accordance with four standardized phases.

2.1. Literature search strategy

To collect relevant literature in the field of fog computing task scheduling, this study conducted searches across five authoritative databases, including IEEE Xplore. We only included peer-reviewed journal articles and conference papers, limited included publications to the period from 2020 to 2025, and carried out all searches using a core search query and its variants.

$$\begin{aligned}
 & \text{("Fog Computing" OR "Edge Computing")} \\
 & \text{AND} \\
 & \text{("Task Scheduling" OR "Resource Allocation")} \\
 & \text{AND} \\
 & \text{("Internet of Things" OR "IoT")} \\
 & \text{AND} \\
 & \text{("Machine Learning" OR "Reinforcement Learning")}
 \end{aligned} \tag{1}$$

Additional manual searches were performed using references from highly cited survey papers and recent review articles to ensure comprehensive coverage of the literature.

2.2. Inclusion and exclusion criteria

To maintain the quality and relevance of the reviewed studies, predefined inclusion and exclusion criteria were applied.

- a. Inclusion criteria: studies were included if they satisfied the following conditions:
 - Published in peer-reviewed journals or conference proceedings.
 - Written in English.
 - Focused on task scheduling, resource allocation, or workload management in fog or edge computing environments.
 - Addressed IoT-enabled applications or large-scale distributed systems.
 - Proposed, analyzed, or evaluated intelligent scheduling approaches, including heuristic, metaheuristic, ML, RL, or hybrid methods.
- b. Exclusion criteria: studies were excluded if they:
 - Focused exclusively on traditional cloud computing without fog or edge components.
 - Did not address scheduling-related problems.
 - Were duplicate publications.
 - Were editorials, tutorials, book chapters, theses, or non-peer-reviewed reports.
 - Lacked sufficient technical details or evaluation results.

2.3. Study selection process

This study conducted a systematic review. After completing database searches, duplicate removal, preliminary screening, and full-text assessment, we included all eligible literature, and built an analytical framework centered on core elements such as scheduling paradigms and optimization objectives. Figure 1 illustrates the overall literature selection process.

The literature review process adopted in this study is divided into four core stages: identification, screening, eligibility assessment, and final inclusion. All deduplication operations conducted during data collection in the identification link of the first stage, as well as all quantitative data, have been fully retained to guarantee the replicability of this study.

This study is currently in the literature screening stage. We have screened the titles and abstracts of 421 original studies, and excluded 287 records that do not cover the four core research areas (including task scheduling) in accordance with pre-set criteria. Following the initial screening process, this study conducted eligibility assessments of the full texts of 134 articles, and excluded 56 articles that did not meet the inclusion criteria; the reasons for these exclusions are divided into four categories.

This review ultimately included 78 studies that met all screening criteria. All studies were categorized by scheduling paradigm, optimization objective, intelligent dimension, scalability characteristic, and safety consideration. The dataset generated from this process supports the subsequent construction of the classification system, comparative analysis, and identification of research gaps.

2.4. Classification framework

The selected studies were systematically classified according to the following dimensions:

- Scheduling paradigm (heuristic, metaheuristic, ML, RL, and hybrid approaches).
- Adaptability (static or dynamic scheduling).
- Optimization objectives (latency, energy efficiency, scalability, cost, and reliability).
- Intelligence dimensions (priority-awareness, security-awareness, and locality-awareness).

- Computational complexity and real-time suitability.

This classification framework enables a comprehensive comparison of existing intelligent scheduling models while identifying research gaps and future opportunities for fog-enabled IoT systems.

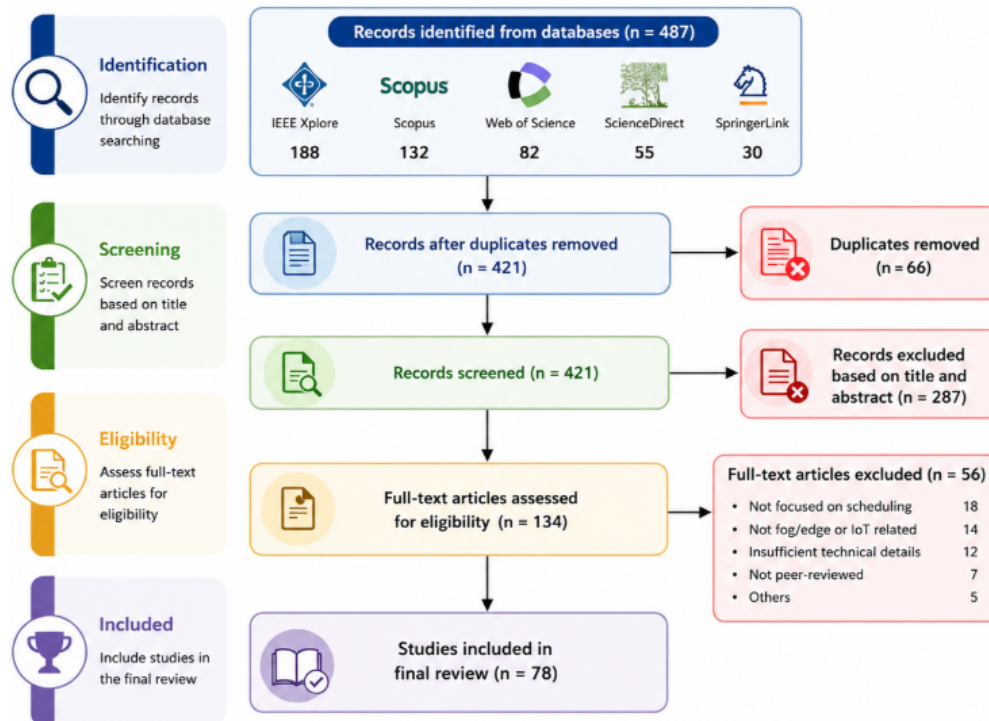


Figure 1. PRISMA-based literature selection process used in this survey

3. FOG COMPUTING AND SCHEDULING FUNDAMENTALS

Fog computing has gained an increasing attention as the central architectural paradigm to overcome the drawbacks of cloud-centric processing for massive IoT systems. With computations, storage, and networking at or near the data sources, fog computing provides low-latency processing (delay-sensitivity), higher scale physical devices able to meet user demand for scale up/down resources as needed and increased context awareness. This section briefly reviews fog computing basics and presents the importance of task scheduling as a fundamental tool for optimal resource management in fog-dependent IoT model.

3.1. Fog computing architecture in internet of things

As shown in Figure 2, overall structure of a conventional fog computing system consists of three logical layers: device, fog, and cloud. The device layer involves IoT sensors, actuators, and smart devices that produce data and execute computational operations. These devices have limited resource availability and would require aid from neighboring fog nodes for processing support. The fog is composed of individual fog nodes that include gateways, routers, micro data centers, and edge servers equipped with intermediate computation and storage capabilities. With the fog layer performing computation near data sources, communication delay, and network traffic are alleviated. The cloud tier provides centralized processing and computational storage for computation-heavy workloads and worldwide analytics that are not latency-sensitive.

Unlike edge computing, fog computing enables multi-hop processing and coordination over multiple intermediate nodes rather than simply at the network edge. This layered structure enables fog systems to accommodate a wide-range of IoT applications that have different latency, bandwidth, and reliability constraints.

3.2. Task scheduling in fog computing

Task scheduling in fog computing is the act of allocating computational tasks to suitable executor nodes over the fog and cloud layer, as shown in Figure 3. Unlike cloud scheduling, fog scheduling is subject to

highly dynamic environments such as dynamically varying workloads, diverse capabilities among nodes, volatile network connection and mobility of devices and users. Efficient scheduling decisions have a direct effect on system performance characteristics, response time throughput, energy consumption, and QoS [12], [14].

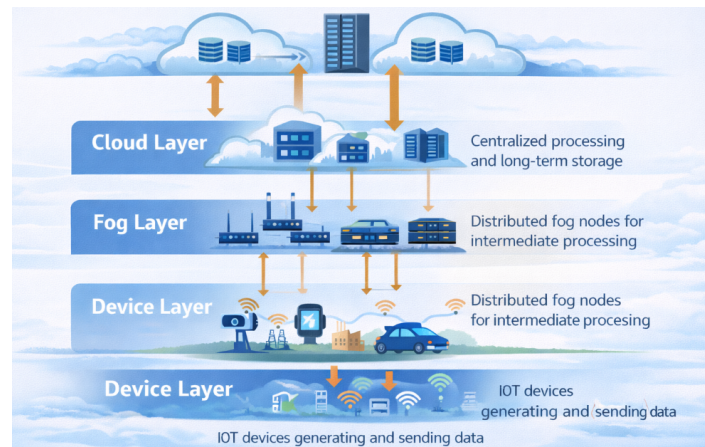


Figure 2. Three-layer fog computing architecture for IoT environments



Figure 3. Schematic representation of task scheduling in fog computing

When scheduling tasks in the fog, schedulers usually take a number of factors into account such as task urgency, availability of resources and network latency and execution cost. According to application demands, tasks may be processed on fog nodes, migrated among different fog regions and offloaded into the cloud [15], [16]. The inefficient scheduling can cause high latency, low resource utilization and even the service degradations especially in large-scale IoT [17], [18].

3.3. Scheduling objectives and performance metrics

The aim of fog scheduling is to meet the requirements of task deadline with effective resource utilization in distributed environment [12]. Typical scheduling objectives include the minimization of end-to-end latency, task waiting and completion time as well as better load balancing between fog nodes and energy consumption [19]. Furthermore, the contemporary fog schedulers are increasingly involving fairness, reliability and security-awareness objectives into their models.

Authors commonly use performance measures including response time, makespan, resource utilization, energy consumption, level of satisfaction in task execution and scheduler overhead to access the scheduling efficiency [20]. In a macro-IoT system, these schedulers also need to be evaluated in terms of scalability and adaptability since they should remain efficient when task arrival rates are large and the communication environment is dynamic [21].

3.4. Difficulties of fog-scheduling for massive internet of things

Scheduling in fog computing encounters various intrinsic challenges, because of its decentralized and heterogeneous characteristics. Resource availability and fog node heterogeneity make the challenge of placing tasks increasingly difficult, while the dynamic nature of networking conditions and nodes' mobility bring uncertainty and instability [13], [22]. Latency-constraint and mission-critical IoT applications require the performance to be enforced within strict time bounds, which is hard to achieve in a consistent manner. Besides, the overhead caused by security and privacy constraints can also be detrimental to scheduling performance [23].

These challenges naturally motivate the design of intelligent scheduling models, which respond to variable conditions, trade off competing objectives and act efficiently at internet scale. In the rest of this paper, such intelligent task scheduling models are reviewed and analyzed, their advantages/disadvantages analyzed and applicability in large-scale IoT fog computing system.

4. TAXONOMY DIMENSIONS AND CLASSIFICATION CRITERIA

Fog computing has been proposed as a fundamental architectural paradigm to overcome the deficiencies of cloud-centric processing in massive internet of things (mIoT) deployments. Fog computing achieves locally computationally intensive operations, data storage and networking near data sources to process at low latency, grow in scalability and augment context awareness. This section presents fundamental concepts of fog computing and the notion that task scheduling is identified as a core technique to efficiently manage resources in fog-oriented IoT-based environments.

We take the following dimensions to consistently categorize the literature.

- Decision mechanism: deterministic rule sets, search-based or gradient optimization based, inference of behavior from observed data (data-driven), and policy learning.
- Adaptability: static policies vs. dynamic policies based on workload and the variation of mobility and resources.
- Optimization objectives: single-objective (ex. latency) vs. multi-objective (vs. trade of between latency and energy and cost).
- System knowledge: information regarding network locality, regional constraints, security overhead, and system diversity.
- Computation overhead: whether scheduling in real-time is feasible due to resource scarcity of the fog.

According to these dimensions, intelligent task scheduling solutions can be classified into five primary types: heuristic and rule-based schemes, metaheuristic schemes, ML mechanisms, RL device, hybrid or adaptive mechanism.

4.1. Heuristic and rule-based scheduling models

Heuristic and rule-based scheduling models have been the first and most popular category of task schedulers in fog computing environments for their simplicity, less computation, and easy implementation [24], [25]. Note that these models either based on predened rules, static priorities or light-weight decision functions used to assign tasks into available fog nodes, as shown in Figure 4. Typical scheduling algorithms include, but are not limited to, First-In First-Out (FIFO), Shortest Job First (SJF), Earliest Deadline First (EDF), Round Robin (RR), and static priority-based schedulers [26]. In fog-based of IoT systems, such mechanism may be enriched with very basic context data, like node availability, queue length or distance to the data source in order to minimize latency and communication overhead [27], [28].

The advantage of heuristic and rule-based schedulers is their speed in decision-making as the timely scheduling in resource-limited fog nodes for latency-sensitive applications are crucial [29]. Its deterministic properties provide predictable system behaviour and simple integration with container orchestration platforms and compact fog middleware. Therefore, these schedulers are often taken as basic polices or building blocks in other late schedulings [30].

However, heuristics and rule-based models have poor performance in large-scale and dynamic IoT networks. Static logic limits all adaptations to workload changes, mobile nodes, and the heterogeneous resources. Another issue with priority-based heuristics is that they naturally lead to task starvation or can be defined as low-priority tasks starving for servicing time in a high prioritization environment. These models also fail to handle multi-objective optimization like minimizing latency, energy, and security overhead in parallel. Although heuristic and rule-based scheduling models are attractive in simple or resource constrained environments, they fall short under the complexity and scale requirements of today fogs paradigms.

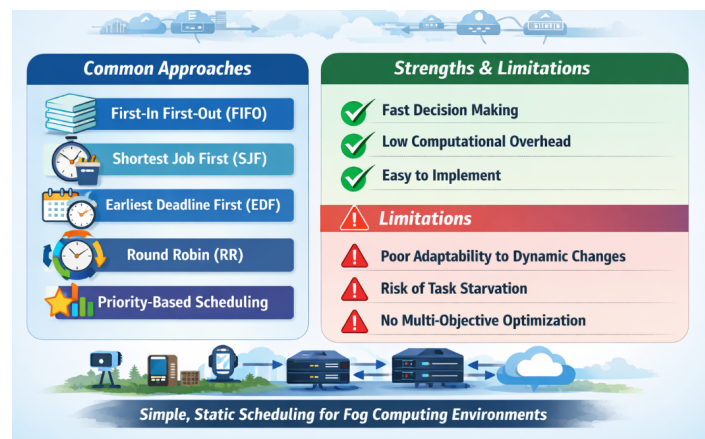


Figure 4. Overview of heuristic and rule-based task scheduling models in fog computing

4.2. Metaheuristic scheduling models

Due complexity and multi-objectiveness of task scheduling in large -scale IoT environments, metaheuristic scheduling models have been heavily applied in fog computing. Unlike heuristic-based solutions that use fixed rules, metaheuristic models employ population or evolutionary search strategies to extend into a wider solution space and provide close-to-optimal scheduling decisions, as shown in Figure 5. This can often be achieved by means of metaheuristic algorithms like genetic algorithms (GA), particle swarm optimization (PSO), ant colony optimization (ACO), differential evolution (DE), grey wolf optimization (GWO), and other bio-inspired methods.

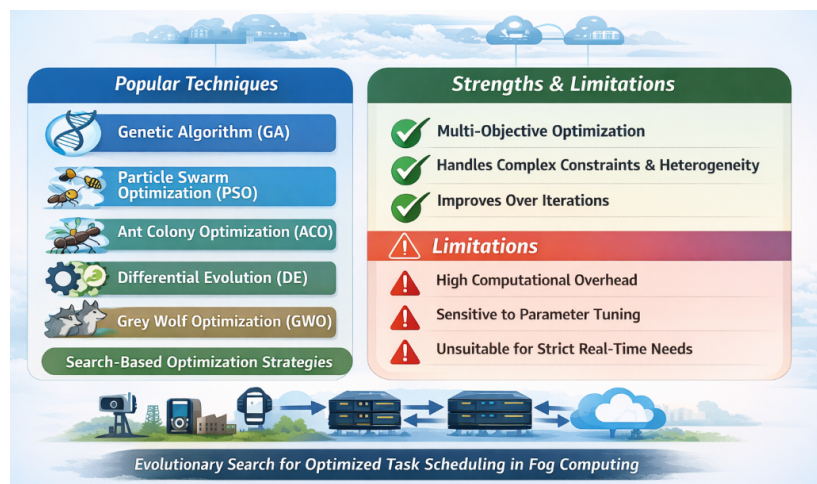


Figure 5. Conceptual overview of metaheuristic scheduling models in fog computing

In fog systems, metaheuristic schedulers are mainly developed in order to decide multiple and conflicting objectives (latency and completion time task minimization versus energy consumption reduction or workload balance between different fog nodes or operational savings). They are flexible enough to include a variety of constraints such as node heterogeneity, limited network bandwidth, mobility, and task deadlines. Hence, metaheuristic approaches have generally provided better solutions than classical heuristics in complex scheduling problems, especially when global optimization is involved.

Despite their effectiveness, however, metaheuristic scheduling models are certainly faced with major obstacles in realistic fog scenarios. This takes a higher computational cost due to the iterative search and evaluation approach that is based on population which is impractical for applications with tight latency requirements or fog nodes that have limited capabilities. Performance is also susceptible to the parameterization, notably the population size, mutation rate or convergence thresholds, that usually have to be extensively hand-tuned. As a result, although metaheuristic scheduling optimization models provide powerful solutions to the complex fog

systems optimization problems, they may need to be simplified or hybridized with simpler heuristics or even offloaded to more capable devices in order ensure the scalability and responsiveness of actual deployments.

4.3. Machine learning-based scheduling models

Fog computing has paid considerable attention to ML-based scheduling models, in order to address the limitation of static and rule-based schedulers in large-scale IoT systems, in Figure 6. Nowadays the abundance of finegrain system activity data can be exploited by learning models to infer scheduling decisions from historic system behavior, which can help achieve proactive and adaptive task placements under dynamic workload and network factors [31], [32]. Contrary to heuristics or metaheuristics depending on pre-selected rules or iterative search, ML-based schedulers predict system's relevant parameters and act based on this expectation for scheduling [33].

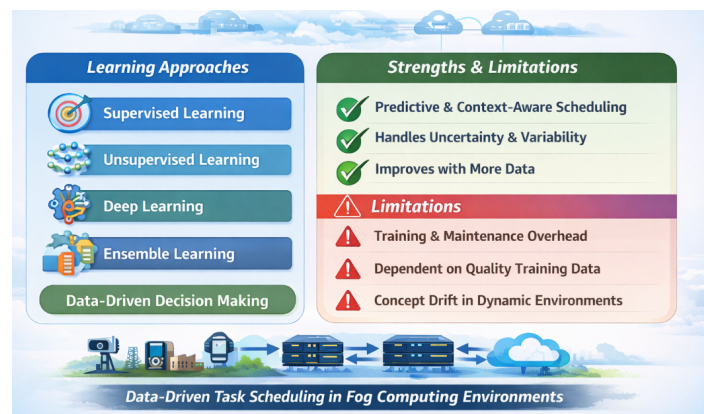


Figure 6. Conceptual overview of ML-based scheduling models in fog computing

Task execution time, network delay, energy consumption, or node availability are often estimated based on past observations using supervised learning methods. Once having the ranks, we construct regression and classification models to rank candidate fog nodes or choose execution places against predefined objectives. Unsupervised learning techniques (e.g., clustering) are usually incorporated to cluster the fog nodes by resource similarity, geographic proximity, or workload patterns with a goal of reducing decision complexity in addition to enhancing scalability. Ensemble learning and deep learning methods are also proposed to model the non-linear relationships between system states and scheduling results in some studies.

The primary advantage of ML-based scheduling models is generalization from data, supporting predictive context-aware decision making. This allows better handling of uncertainty, workload diversification and heterogeneity among fog nodes. But they are only as good as the data on which they train. There may also be intensive computation overhead associated with model training and updates for resource-limited fog environments. In addition, performance can deteriorate because of concept shift, brought about when the driving conditions are far from the ones encountered when training data was collected. Consequently, although ML scheduling models possess great flexibility and prediction ability, designing light-weight models, retraining model periodically or collaborating with simpler scheduling schemes must be employed to guarantee the robustness and real-time feasibility.

4.4. Reinforcement learning scheduling models

RL [34], [35] based scheduling models have been identified as a potential candidate due to their superior intelligence compared to rule-based schedulers for fog computing systems, which are commonly found in large-scale IoT networks that exhibit dynamic workloads and resources, and fluctuating network states, as shown in Figure 7. In contrast to ML-based supervised approaches which are based on labeled historical data, an RL-based scheduler learns optimal scheduling policies by interacting directly with the environment observing system states, taking actions and receiving feedback in terms of rewards.

In fog computing, RL has been widely employed to formulate the task scheduling as a sequential decision making problem. What is more, state space can consist of; i) fog nodes utilisation, ii) queue lengths, iii) network delay, and iv) task urgency or energy amount and actions may mean choosing execution/offloading

destinations. Traditional RL algorithms like Q-learning have been used in the contexts of low-dimensional cases, while deep reinforcement learning (DRL) approaches (e.g., deep Q-networks (DQN), proximal policy optimization (PPO), and actor-critic models) are being used more often to address high-dimensional state and action spaces.

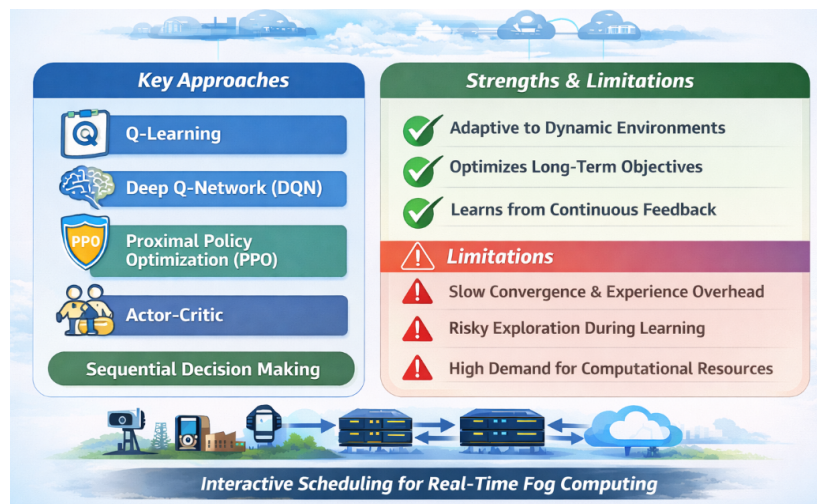


Figure 7. Conceptual overview of RL-based scheduling models in fog computing

One of most attractive characteristics when dealing with RL-based scheduling approach is its high versatility and effectiveness in terms of long-run optimization. RL schedulers can adapt by continually learning new policies to react to workload variation, node movement, and network condition changes. This makes it suitable for real-time or mission-critical IoT applications in which static or predictive models may not generalize.

Yet, RL-based scheduling method also comes with various difficulties. Training and convergence could be slow, particularly in large-scale fog environment with complicated state spaces. Learning-based exploration may infer suboptimal or unsafe scheduling policies, which would be undesirable for real-time critical safety-related tasks. Furthermore, the computational cost to train deep models could be beyond the capacity of resource-limited fog nodes. As a result, practical RL-based schedulers often rely on centralized or hierarchical learning reward shaping, and/or with more naive heuristic components to achieve stability, scalability and applicability in real-world scenarios.

4.5. Hybrid and adaptive scheduling models

Hybrid and adaptive scheduling mechanisms are part of a new generation of smart schedulers developed to overcome the disadvantages of single-paradigm model in fog computing, as shown in Figure 8. These frameworks combine a variety of scheduling methods (e.g., heuristics, metaheuristics, ML, and RL) to compromise among the decision quality, adaptability, and computational effort in the context of large-scale IoT systems. Hybrid methods are primarily born out of the motivation to benefit from conjunction of two approaches with complementary advantages and counterpart disadvantage.

In the fog computing domain, heterogeneity scheduling often combines lightweight heuristics with learning-based or optimization-driven methods. For example, initial task filtering or fast decision making can be achieved by heuristic rules and placement decisions are further refined for optimization by metaheuristic or learning-based modules. In other related works, ML models are used for parameter prediction in the system-side and RL is performed to dynamically adjust scheduling policies over time. Adaptive scheduling models provide greater integration by integrating feedback-based, threshold-based triggers, or periodic policy updates that enable the scheduler to respond to workload variability, node physical motion and network dynamics.

The key benefit of the hybrid and adaptive scheduling is robustness and adaptability. When scheduling behaviour is dynamic and adapted to real-time system conditions, these models can result in better latency, resource usage, fairness and scalability than with static or single-method schedulers. This renders them well-suited for heterogeneous and non-stationary fog networks that serve mission-critical IoT applications.

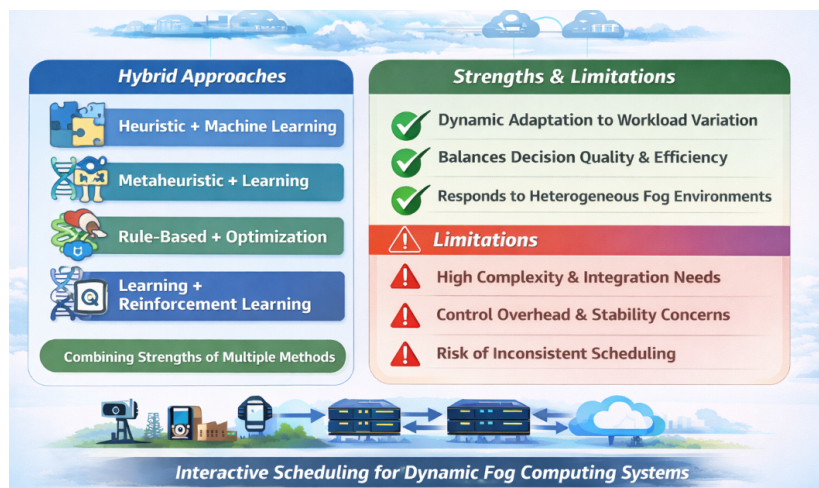


Figure 8. Illustration of hybrid and adaptive task scheduling approaches in fog computing

However, the design complexity of hybrid and adaptive models has become significantly convoluted. In order to integrate multiple scheduler components, an in-depth coordination, parameter optimisation and picture for consistency management are necessary in order to prevent interference between decision making. Furthermore, it is likely that adaptive procedures may lead to control overhead and stability issues if they are not properly designed. Therefore, although hybrid and adaptive scheduling schemes are seen as a promising trend in the future fog systems, their realistic deployment requires well consideration on architectural design and thorough evaluation to guarantee real-time feasibility and system reliability.

4.6. Cross-cutting scheduling intelligence framework

Recent years have witnessed the latest research progress in the field of fog computing scheduling, which shows that scheduling intelligence is no longer determined solely by a single underlying optimization algorithm. Effective decision-making must rely on cross-paradigm, interdisciplinary intelligence dimensions, which can cover all mainstream schedulers across all major categories: heuristic, meta-heuristic, ML, RL, and hybrid approaches.

This paper combed through published review literature, and extracted three cross-domain intelligent dimensions: priority, safety, and location awareness, to supplement the four optimization goals of traditional technologies. To formalize these observations, this survey proposes the cross-cutting scheduling intelligence framework illustrated in Figure 9. The framework integrates multiple intelligence dimensions into a unified scheduling decision process for fog-enabled IoT environments.

4.6.1. Priority awareness

Priority-aware is an important aspects of intelligence, especially for fog scheduling in IoT systems with mixed timing-sensitive and QoS-sensitive applications. A number of scheduling systems have introduced task priority schemes to distinguish between delay-sensitive, safety-critical, and best-effort workloads. Most of these priority-adaptive schedulers preempt when high-priority tasks are ready to be run, enforce processing deadlines for jobs, or use weighted queues to ensure that high-importance work is performed without undue delay. Nevertheless, if priorities are not managed properly, low-priority tasks might be starved of resources—indicating the requirement to fairness-aware or adaptive priority adjustments for large-scale deployments.

4.6.2. Security awareness

Although regarding fog in general as the extension and decentralization of cloud computing which is still vulnerable to many threats such as much high execution tasks and IoT capture enriched other, in recent years security awareness for this setting has become more serious. Scheduling will not take into account security overhead introduced for example by encryption, authentication, trusted execution environment or isolation mechanisms. Security-conscious schedulers attempt to optimize performance goals while respecting protection needs through dynamic node selection (based on security capabilities) or adjusting levels of security per task sensitivity. However, with increasing popularity there is still limited attention in the existing scheduling models

to characterize and manage security awareness including level of security concerns and overhead (latency or resource penalty) caused by operating at a certain security-level.

Locality and region awareness strategies target minimal communication latency and network congestion by scheduling tasks as close as possible to their data source (locality) or within a particular local area, such as in an administration domain/region. In ultra-scale IoT systems, proximity based scheduling is especially important for supporting mobility, alleviating backhaul traffic, and ensuring service continuity. For instance, region-aware schedulers implement clustering as well as zoning or hierarchical control structures in order to accommodate dynamic clustering or shifting scheduling (among others).

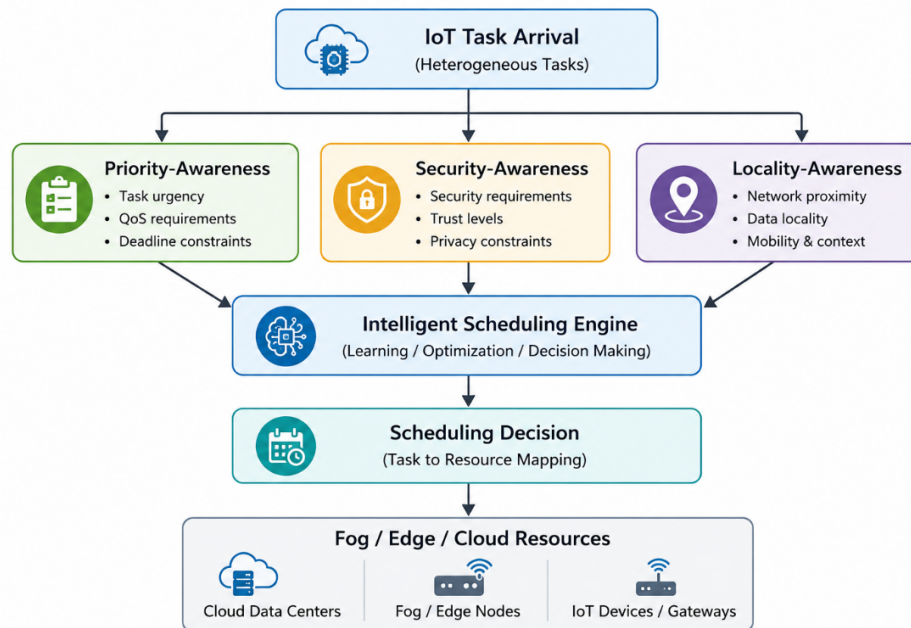


Figure 9. Illustration of cross-cutting intelligence features commonly integrated into fog computing schedulers

5. PRIORITY-AWARE SCHEDULING IN FOG COMPUTING

Priority aware scheduling is one of the challenges in fog computing, as tasks have different levels of urgency and latency requirement depends on various classes priority basic temperature background normalised function most with span step realisation imply confuses definitely at least mister [36], [37]. In contrast to merely best-effort based scheduling schemes, priority-aware schedulers segregate tasks based on their importance, and QoS needs. This section presents the work-flow and design issues of the priority-aware scheduling procedure step-wise [38], [39]. Figure 10 illustrates the workflow of a priority-aware scheduling framework in fog computing environments.

The framework incorporates task prioritization, resource assessment, priority-based task selection, dynamic priority adjustment, and fairness enforcement mechanisms to support heterogeneous IoT applications with diverse QoS requirements.

- Characterization task and priority assignment: first, classify the series again according to its application and service requirements. Based on deadline constraint, task urgency and nature of application, you get to assign priority levels. These classes can be a safety-critical, latency-critical and background tasks that allow the run to wait more.
- Priority classification and queuing: following the priority assignment, tasks are queued into appropriate queues or weighted priority stacks. Preemption is supported in the high priority queues, while non-preemptive (i.e., best-effort) scheduling is done inside low-priority queues. This classification offers scheduler per-queue isolation and differential service guarantee.
- Every certain period of time, the scheduler checks states of the resources on fog nodes like CPU load, available memory and network to assess resource availability. This assessment ensures that priority decisions are made using the latest system capacity, and prevents overloading of some fog nodes from hyperservicing high-priority workloads.

- Priority based selection of task: jobs have their task selection depending upon the priority and state of queue. High-priority jobs will be scheduled ahead of low-priority ones, almost always preempting a low priority job if work from high priority queue comes in. Aging mechanisms can be built into the selection policy that slowly elevate the importance of jobs waiting for a long time so that starvation does not occur.
- Execution and monitoring: some processes are dispatched to relevant fog nodes for execution progress and performance information (including response time, deadline of execution) are monitored at runtime. Monitoring helps catch missed deadlines or degradation in performance early on.
- Dynamic priority adjustment: dynamic priority-aware schedulers adjust the priorities dynamically to adapt the system at changing load or context and with respect to actual observed execution behavior. For example, jobs that are nearing their deadline can be temporarily increased in priority to make sure that gets completed on time.
- Feedback and fairness enforcement: finally, completed tasks provide feedback to refine the priority policy as well as to balance classes of different types of tasks. To balance responsiveness with respect to high-priority jobs and fairness in resource allocation with respect to lower-priority workloads, we use preemption and starvation-prevention mechanisms such as aging-based priorities or quota-based scheduling.

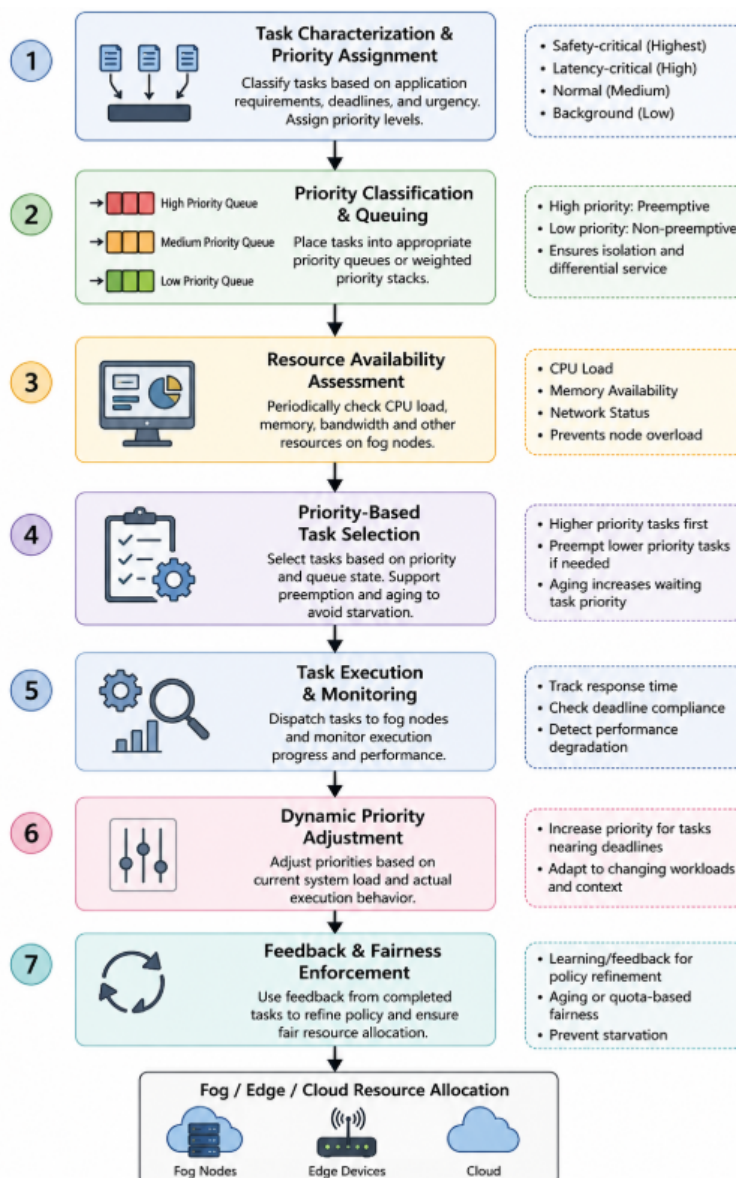


Figure 10. Workflow of priority-aware task scheduling in fog computing environments

With the above steps, the fog computing system can support heterogeneous IoT applications with diversity of QoS requirements using priority-aware scheduling frameworks. Though it harvests the maximum throughput of key flows by efficiently using priority-aware scheduling, we should equip such a design with meticulous trade-offs smoothing responsiveness.

6. COMPARATIVE ANALYSIS AND DISCUSSION

This subsection surveys a overview of intelligent activity scheduling fashions in fog computing for large-scale IoT structures. The goal here is not to enumerate a single best solution but rather, demonstrate how various scheduling paradigms find a balance between these four metrics: adaptability, computational overhead, scalability, and real-time readiness as shown in Table 2. This benchmark is necessary for understanding the actual gain achieved by utilizing scheduling intelligence in heterogeneous and dynamic fog systems.

Table 2. Comparative analysis of intelligent task scheduling models in fog computing for large-scale IoT networks

Scheduling model	Adaptability	Computational overhead	Scalability	Real-time suitability	Key strengths and limitations
Heuristic and rule-based	Low	Very low	Moderate	High	Fast decision making and low overhead; limited adaptability and poor support for multi-objective optimization.
Metaheuristic	Medium	High	Moderate	Low	Effective for multi-objective optimization and complex constraints; high computational cost and parameter sensitivity limit real-time deployment.
ML-based	Medium to high	Medium	High	Medium	Predictive and data-driven scheduling; performance depends on data quality and suffers from concept drift in dynamic environments.
RL-based	High	High	High	Low to medium	Strong adaptability and long-term optimization; slow convergence and exploration overhead pose challenges for safety-critical tasks.
Hybrid and adaptive	Very high	Medium to high	Very high	Medium	Balances performance and adaptability by integrating multiple paradigms; increased design complexity and coordination overhead.

The quantitative trends summarized in Table 3 provide a high-level comparison of intelligent scheduling paradigms reported in the literature. This paper reviews four mainstream scheduling technologies, all of which are compared along the dimensions of computational cost and performance gain: i) the heuristic rule-based method features low overhead and strong real-time performance, but has a limited performance ceiling; ii) metaheuristic methods have strong optimization capabilities but high complexity; iii) ML-based methods can reduce latency and improve energy efficiency; and iv) the hybrid framework delivers the best overall performance, supporting the intelligent development trend of large-scale fog-enabled IoT integrated scheduling.

Table 3. Quantitative performance trends reported by intelligent scheduling approaches in fog computing

Scheduling paradigm	Latency reduction (%)	Energy saving (%)	Scalability	Real-time suitability
Heuristic and rule-based	10–20	5–15	Moderate	High
Metaheuristic	25–40	20–35	Moderate	Low
ML-based	30–50	25–40	High	Medium
RL-based	35–60	30–45	High	Medium
Hybrid and adaptive	40–70	35–55	Very high	Medium

6.1. Intelligent scheduling for embedded and reconfigurable fog systems

The deployment scale of fog computing infrastructure continues to expand, creating new opportunities for integrating intelligent task scheduling mechanisms into embedded reconfigurable computing platforms. Compared with traditional cloud environments, fog nodes face four categories of resource constraints, so any appropriately adapted scheduling mechanism must simultaneously meet the dual core standards of performance optimization and its own high-efficiency deployability.

At present, the typical carriers of embedded fog nodes include IoT gateways, edge servers, industrial controllers, and smart routers. These devices are universally integrated with intelligent scheduling modules to support real-time decision-making [40]. Lightweight heuristic schedulers are widely adopted due to their low

computational overhead; while ML-based solutions can adapt to dynamic load fluctuations, they are constrained by limits of memory and complexity, making them difficult to deploy on embedded devices [41].

Reconfigurable computing platforms, particularly field-programmable gate arrays (FPGAs), are highly promising solutions for accelerating scheduling operations in fog environments [42], [43]. Compared with pure software implementation schemes, these platforms can execute three core types of tasks, achieve lower latency and reduced energy consumption, and support real-time scheduling decisions for four key IoT applications [44], [45].

This paper review some hardware options for intelligent scheduling engines excluding FPGAs, such as application-specific integrated circuits (ASICs), low-power very-large-scale integration (VLSIs), and other relevant architectures. These hardware platforms must meet strict energy consumption limits, and are capable of supporting parallel task evaluation and resource matching [46]. The paper further proposes that hardware-software co-design forms the core of scalable, high-responsiveness fog computing infrastructure [47], [48].

This paper proposes that the future scheduling framework for fog-enabled IoT must balance algorithmic intelligence and hardware implementation. This framework will integrate three categories of algorithms—ML, RL, and hybrid scheduling—and two types of hardware platforms, namely embedded and reconfigurable platforms, to build a low-latency, energy-efficient, and scalable system. As shown in Figure 11, intelligent scheduling modules can be integrated into embedded processors, FPGA-based accelerators, and edge gateways to support efficient task allocation in fog-enabled IoT environments.

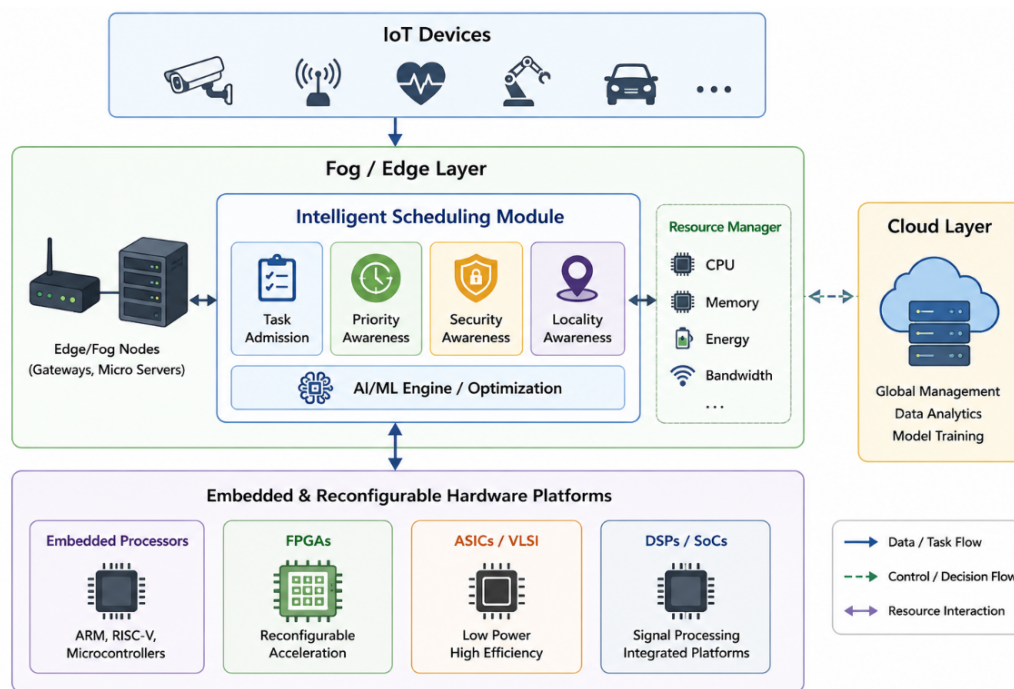


Figure 11. Integration of intelligent scheduling modules within embedded and reconfigurable fog computing platforms

Table 4 presents the potential benefits of intelligent scheduling when deployed on embedded and reconfigurable fog computing platforms, highlighting their suitability for real-time and resource-constrained IoT environments.

Table 4. Role of intelligent scheduling in embedded and reconfigurable fog systems.

Platform	Scheduling benefits
IoT gateway	Low-latency task allocation
Embedded processor	Lightweight scheduling decisions
FPGA	Hardware acceleration and parallel scheduling
ASIC/VLSI	Ultra-low-power scheduling execution
Edge server	Adaptive AI-driven scheduling

7. FUTURE DIRECTIONS AND OPEN CHALLENGES

Although remarkable progress has been made in intelligent task scheduling within the field of fog computing, challenges still persist in large-scale IoT environments that have heterogeneous resources, dynamic workloads, and stringent quality of service requirements. Resolving these challenges is the core prerequisite for developing scalable scheduling frameworks.

- Scalability in large-scale IoT environments: with the rapid growth of IoT devices, the influx of large-scale tasks has raised scheduling complexity. Existing centralized scheduling models suffer from the drawbacks of insufficient scalability and excessively high coordination overhead, making it necessary to develop hierarchical, distributed, and decentralized scheduling architectures that meet the requirements of large-scale deployment, low latency, and high resource utilization.
- Resource heterogeneity and mobility awareness: heterogeneous devices in fog environments feature uneven computing capacity, energy constraints, and varying mobility. Their intelligent scheduling frameworks must adapt to dynamic changes, and context- and mobility-aware strategies can improve service continuity and resource utilization.
- Real-time responsiveness and lightweight intelligence: although ML and RL have the core advantage of adaptive decision-making, their excessively high computational complexity hinders the deployment of scheduling systems in real-time application scenarios. This paper proposes three types of optimization solutions: lightweight learning models, model compression technologies, and hybrid architectures that integrate heuristic decision-making.
- Security and privacy-aware scheduling: this paper propose that the vast majority of existing scheduling schemes only focus on performance optimization and fail to address security and privacy requirements. Future schedulers must embed dynamic security awareness capabilities and incorporate five categories of security constraints.
- Stability and safety of learning-based scheduling: existing RL-based schedulers exhibit three core flaws, including slow convergence and unstable behavior, and five tailored technical improvement directions can be advanced for mission-critical IoT systems.
- Benchmark standardization and real-world validation: this paper propose that the research field of scheduling methods lacks standardized datasets, benchmarking frameworks, and realistic testbeds, which makes it impossible to fairly compare different scheduling schemes, and that four core research directions must be advanced.
- Cross-layer and multi-dimensional scheduling intelligence: this paper proposes that current mainstream scheduling algorithms can only optimize a limited set of objectives, and future frameworks must adopt a unified cross-layer optimization strategy to balance and meet all six core requirements.
- Edge artificial intelligence (edge AI): this paper proposes that edge AI can support fog nodes and edge nodes to complete intelligent decision-making directly. Relying on lightweight edge deep learning models, the future network scheduling framework realizes real-time task allocation, workload prediction and adaptive resource management, while minimizing communication overhead.
- Federated learning-based scheduling: federated learning enables distributed fog nodes to complete collaborative scheduling optimization without sharing raw data. Future research must develop a privacy-preserving scheduling mechanism for distributed model training that balances both data confidentiality and reduced communication costs.
- Blockchain-assisted scheduling: blockchain technology can enhance trust and transparency in distributed fog environments, and when paired with a scheduling framework, it can support the secure deployment of multi-domain IoT systems.
- Digital twin-enabled scheduling: we propose that digital twins can generate virtual replicas of physical systems to support predictive scheduling and resource optimization. When incorporated into a fog scheduling framework, digital twins can predict workloads, evaluate operational strategies, and improve system resilience.
- Explainable artificial intelligence (XAI) for scheduling: scheduling-oriented XAI is the research track defined in this paper. It addresses the core pain point of rising complexity of scheduling mechanisms, is adapted for safety-critical IoT applications, and can improve the transparency and credibility of decision-making.
- Hardware-accelerated scheduling for embedded and reconfigurable systems: this paper propose that future intelligent schedulers for embedded fog platforms must adopt three types of dedicated acceleration architectures, improve efficiency through hardware-software co-design, and simultaneously meet strict constraints on energy consumption and latency.

Overall, future intelligent scheduling frameworks are expected to evolve toward unified, adaptive, secure, and hardware-aware solutions that integrate advanced AI techniques with scalable fog computing infrastructures. Bridging the gap between theoretical scheduling models and practical deployments will require stronger emphasis on real-world validation, security-awareness, explainability, and embedded implementation considerations.

8. CONCLUSION

This paper focuses on fog computing scenarios within large-scale IoT environments, presents a comprehensive review of intelligent task scheduling models, categorizes prior research into five major scheduling paradigms, and conducts a cross-paradigm comparison across four types of dimensions, including adaptability and computational overhead. Previous surveys on scheduling for fog computing-supported IoT systems only focused on algorithm classification and had a single limited research perspective. This survey introduces a multi-dimensional intelligent analysis perspective for cross-domain scheduling, breaks down three core dimensions: priority, security, and location awareness, sorts out the trade-offs between the strengths and weaknesses of various scheduling paradigms, and reaches the conclusion that no universal optimal scheduling strategy exists that is suitable for all scenarios. This study compares two categories of scheduling technologies. Heuristic methods remain valuable for their adaptability to high real-time scenarios with constrained resources. ML-based methods boast strong adaptability and enable long-term optimization. A hybrid adaptive framework can integrate the respective advantages of these two types of methods while avoiding their weaknesses, making it the optimal development direction. This paper have sorted out and summarized all some open challenges and emerging research directions in the field of fog computing. These directions will play an important role in the development of the next-generation intelligent scheduling framework for secure, scalable, and adaptive fog computing infrastructures.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Sababathy Palanisamy	✓	✓		✓	✓	✓		✓	✓	✓				✓
Erum Ashraf		✓		✓	✓	✓		✓	✓	✓				
Selvakumar Manickam	✓	✓		✓	✓		✓			✓		✓		✓
Huda Mohammed Alsayednoor				✓	✓	✓		✓	✓	✓	✓			
Mahmood A. Al-Shareeda	✓	✓		✓	✓	✓		✓	✓	✓	✓			
Shankar Karuppayah		✓		✓	✓		✓			✓		✓		

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal Analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project Administration

Fu : Funding Acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that no funding was received for this study.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

- [1] K. C. Rath, A. Khang, and D. Roy, "The role of Internet of Things (IoT) technology in Industry 4.0 economy," *Advanced IoT technologies and applications in the industry 4.0 digital economy*, CRC Press, pp. 1–28, 2024.
- [2] S. Rekha, L. Thirupathi, S. Renikunta, and R. Gangula, "Study of security issues and solutions in Internet of Things (IoT)," in *Materials Today: Proceedings*, 2023, vol. 80, pp. 3554–3559, doi: 10.1016/j.matpr.2021.07.295.
- [3] M. Dhanaraju, P. Chenniappan, K. Ramalingam, S. Pazhanivelan, and R. Kaliaperumal, "Smart farming: Internet of Things (IoT)-based sustainable agriculture," *Agriculture*, vol. 12, no. 10, p. 1745, 2022, doi: 10.3390/agriculture12101745.
- [4] M. M. Qasim, M. Ahmad, M. Omar, A. N. Zulkifli, and J. Bakar, "Persuasive technology and mobile healthcare: A critical review," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 10, no. 10, pp. 1501–1513, 2018.
- [5] K. Sallam, M. Mohamed, and A. W. Mohamed, "Internet of Things (IoT) in supply chain management: challenges, opportunities, and best practices," *Sustainable Machine Intelligence*, vol. 2, pp. 1–32, 2023, doi: 10.61185/SMIJ.2023.22103.
- [6] M. M. Qasim, M. Ahmad, M. Omar, A. N. Zulkifli, and J. A. A. Bakar, "A systematic process for persuasive mobile healthcare applications," *The 2nd International Conference on Applied Science and Technology 2017 (ICAST'17)*, vol. 1891, no. 1, 2017, doi: 10.1063/1.5005448.
- [7] R. Mishra and A. Mishra, "Current research on Internet of Things (IoT) security protocols: A survey," *Computers & Security*, vol. 151, p. 104310, 2025, doi: 10.1016/j.cose.2024.104310.
- [8] A. Khang and K. Singh, "Internet of Things (IoT) Smart Sensing Traffic Lights for Revolutionizing Urban Traffic Management," *Driving Green Transportation System Through Artificial Intelligence and Automation: Approaches, Technologies and Applications*, Springer, pp. 105–118, 2025, doi: 10.1007/978-3-031-72617-0_6.
- [9] Hemani, D. Singh, and R. K. Dwivedi, "Designing blockchain based secure autonomous vehicular internet of things (IoT) architecture with efficient smart contracts," *International Journal of Information Technology*, vol. 17, no. 2, pp. 1207–1223, 2025, doi:10.1007/s41870-023-01712-x.
- [10] M. M. Qasim, M. Ahmad, and M. Omar, "Persuasive strategies in mobile healthcare: a systematic literature review," *Journal of Engineering and Applied Sciences*, vol. 12, no. 9, pp. 8706–8713, 2017, doi: 10.3923/jeasci.2017.8706.8713.
- [11] W. C. Chuan, S. U. A. Laghari, S. Manickam, E. Ashraf, and S. Karuppayah, "Challenges and Opportunities in Fog Computing Scheduling: A Literature Review," *IEEE Access*, vol. 13, pp. 14702–14726, 2025, doi: 10.1109/ACCESS.2024.3525261.
- [12] B. Jamil, H. Ijaz, M. Shojafar, K. Munir, and R. Buyya, "Resource allocation and task scheduling in fog computing and internet of everything environments: A taxonomy, review, and future directions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–38, 2022, doi: 10.1145/3513002.
- [13] M. Hosseinzadeh *et al.*, "Task scheduling mechanisms for fog computing: a systematic survey," *IEEE Access*, vol. 11, pp. 50994–51017, 2023, doi: 10.1109/ACCESS.2023.3277826.
- [14] A. Najafizadeh, A. Salajegheh, A. M. Rahmani, and A. Sahafi, "Multi-objective Task Scheduling in cloud-fog computing using goal programming approach," *Cluster Computing*, vol. 25, no. 1, pp. 141–165, 2022, doi: 10.1007/s10586-021-03371-8.
- [15] M. Anita *et al.*, "Optimizing task scheduling in fog computing: Adaptive algorithms for enhanced security and efficiency in IoT environments," in *2nd International Interdisciplinary Scientific Conference on Green Energy, Environmental and Renewable Energy, Advanced Materials, and Sustainable Development: ICGRMSD24*, vol. 3193, no. 1, 2024, doi: 10.1063/5.0235647.
- [16] A. M. Yadav, K. N. Tripathi, and S. Sharma, "An enhanced multi-objective fireworks algorithm for task scheduling in fog computing environment," *Cluster Computing*, vol. 25, pp. 983–998, 2022, doi: 10.1007/s10586-021-03481-3.
- [17] M. Hussain, S. Nabi and M. Hussain, "RAPTS: resource aware prioritized task scheduling technique in heterogeneous fog computing environment," *Cluster Computing*, vol. 27, pp. 13353–13377, 2024, doi: 10.1007/s10586-024-04612-2.
- [18] Z. Yin *et al.*, "A multi-objective task scheduling strategy for intelligent production line based on cloud-fog computing," *Sensors*, vol. 22, no. 4, pp. 1555, 2022, doi: 10.3390/s22041555.
- [19] A. Belgacem and K. Beghdad-Bey, "Multi-objective workflow scheduling in cloud computing: trade-off between makespan and cost," *Cluster Computing*, vol. 25, no. 1, pp. 579–595, 2022, doi: 10.1007/s10586-021-03432-y.
- [20] R. Liu, R. Piplani, and C. Toro, "Deep reinforcement learning for dynamic scheduling of a flexible job shop," *International Journal of Production Research*, vol. 60, no. 13, pp. 4049–4069, 2022, doi: 10.1080/00207543.2022.2058432.
- [21] N. Potu, S. Bhukya, C. Jatoth, and P. Parvataneni, "Quality-aware energy efficient scheduling model for fog computing comprised IoT network," *Computers & Electrical Engineering*, vol. 97, p. 107603, 2022, doi: 10.1016/j.compeleceng.2021.107603.
- [22] K. M. Matrouk and A. D. Matrouk, "Mobility aware-task scheduling and virtual fog for offloading in IoT-fog-cloud environment," *Wireless Personal Communications*, vol. 130, no. 2, pp. 801–836, 2023, doi: 10.1007/s11277-023-10310-w.
- [23] M. Gamal, S. Awad, R. F. Abdel-Kader, and K. Abd El Salam, "Efficient offloading and task scheduling in internet of things-cloud-fog environment," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 4, pp. 4445–4455, 2024, doi: 10.11591/ijece.v14i4.pp4445-4455.
- [24] C. Barut, G. Yildirim, and Y. Tatar, "An intelligent and interpretable rule-based metaheuristic approach to task scheduling in cloud systems," *Knowledge-based systems*, vol. 284, p. 111241, 2024, doi: 10.1016/j.knosys.2023.111241.
- [25] W. Liu, J. Zhang, C. Qu, and H. Zhang, "Performance of priority rules for finance-based and resource-constrained project scheduling heuristics," *Journal of Construction Engineering and Management*, vol. 150, p. 04024047, no. 6, 2024, doi: 10.1061/JCEMD4.COENG-14633.
- [26] S. Zhang, X. Song, L. Shen, and L. Xu, "Complicated time-constrained project scheduling problems in water conservancy construction," *Processes*, vol. 11, no. 4, p. 1110, 2023, doi: 10.3390/pr11041110.
- [27] D. Pujara and M. Mathirajan, "Development of Dispatching Rule based Heuristic Algorithms for Real-Time Dynamic Scheduling of Non-identical Parallel Burn-in Ovens with Machine Eligibility Restriction," *International Journal of Mathematical, Engineering & Management Sciences*, vol. 10, no. 3, 2025, doi: 10.33889/IJMEMS.2024.10.3.038.
- [28] İ. O. Çetinkaya, İ. E. Büyüktaktakın, P. Shojaei, and C. K. Reddy, "Discovering heuristics with Large Language Models (LLMs) for mixed-integer programs: Single-machine scheduling," *Computers & Operations Research*, vol. 186, p. 107325, 2025, doi: 10.1016/j.cor.2025.107325.
- [29] L. Becchi *et al.*, "A computationally efficient rule-based scheduling algorithm for battery energy storage systems," *Sustainability*, vol. 16, no. 23, 2024, doi: 10.3390/su162310313.

- [30] F. Zhao, Y. Liu, T. Xu, and Jonrinaldi, "A reinforcement learning hyper-heuristic algorithm for the distributed flowshops scheduling problem under consideration of emergency order insertion," *Applied Soft Computing*, vol. 167, p. 112461, 2024, doi: 10.1016/j.asoc.2024.112461.
- [31] N. K. Gondhi and A. Gupta, "Survey on machine learning based scheduling in cloud computing," in *ISMSI '17: Proceedings of the 2017 International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence*, Mar. 2017, pp. 57-61, doi: 10.1145/3059336.3059352.
- [32] Z. J. K. Abadi, N. Mansouri, and M. M. Javidi, "Deep reinforcement learning-based scheduling in distributed systems: a critical review," *Knowledge and Information Systems*, vol. 66, pp. 5709-5782, 2024, doi: 10.1007/s10115-024-02167-7.
- [33] A. A. A. Kadhim, A. F. Abdulhasan, and F. F. Jaber, "Improvement of extracted photovoltaic power using artificial neural networks MPPT with enhanced flyback controller," *Iraqi Journal for Electrical and Electronic Engineering*, vol. 21, no. 2, pp. 237-250, Dec. 2025, doi: 10.37917/ijeee.21.2.22.
- [34] H. Kang, S. Jung, H. Kim, J. Jeoung, and T. Hong, "Reinforcement learning-based optimal scheduling model of battery energy storage system at the building level," *Renewable and Sustainable Energy Reviews*, vol. 190, p. 114054, Feb. 2024, doi: 10.1016/j.rser.2023.114054.
- [35] H. Lai *et al.*, "On effective scheduling of model-based reinforcement learning," in *Proceedings of the 35th International Conference on Neural Information Processing System*, 2021, pp. 3694-3705, doi: 10.5555/3540261.3540543.
- [36] S. S. Mangalampalli, P. V. Reddy, G. R. Karri, G. Tippi, and H. Kota, "Priority-aware multi-objective task scheduling in fog computing using simulated annealing," *Sensors*, vol. 25, no. 18, p. 5744, 2025, doi: 10.3390/s25185744.
- [37] S. Beborra, S. S. Tripathy, and N. Tripathy, "PriorHealth: a priority-aware task scheduling framework for managing healthcare data in fog computing applications," in *2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, Bengaluru, India, 2024, pp. 586-591, doi: 10.1109/IDCIoT59759.2024.10467807.
- [38] V. Satish, A. R. Kandi, M. P. Kumar, K. Anusha, C. Veena, and S. K. Medishetti, "AAA: Priority-Aware and Latency-Optimized Task Scheduling in Fog Computing," in *2025 4th International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, Salem, India, 2025, pp. 87-94, doi: 10.1109/ICAAIC64647.2025.11331034.
- [39] E. Mashhour, H. A. Khamis, M. Joorabian, S. G. Seifossadat, and S. H. Hammadi, "Emission Reduction for a Home in Iraq Utilizing Geothermal System Supplied by Isolated Photovoltaic-Wind-Diesel Generator Using Genetic Algorithm Optimization," in *2025 Fifth National and the First International Conference on Applied Research in Electrical Engineering (AREE)*, Ahvaz, Iran, Islamic Republic of, 2025, pp. 1-6, doi: 10.1109/AREE63378.2025.10880240.
- [40] M. A. Alhartomi *et al.*, "Low-Area and Low-Power VLSI Architectures for Long Short-Term Memory Networks," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 13, no. 4, pp. 1000-1014, December 2023, doi: 10.1109/JET-CAS.2023.3330428.
- [41] A. Ahmad, S. Hussain, M. A. Raheem, A. Al Maashri, S. S. Al Busaidi, and M. Awadalla, "ASIC vs FPGA based implementations of built-in self-test," *International Journal of Advanced Natural Sciences and Engineering Researches*, vol. 7, no. 6, pp. 14-20, 2023, doi: 10.59287/ijanser.942.
- [42] A. Boutros, A. Arora, and V. Betz, "Field-Programmable Gate Array Architecture for Deep Learning: Survey and Future Directions," *Proceedings of the IEEE*, vol. 113, no. 7, pp. 613-639, Jul. 2025, doi: 10.1109/JPROC.2025.3623023.
- [43] S. Thallal, S. P. Dwivedi, A. Singh, S. L. Dhar, S. Sharma, and N. K. Rajpoot, "Field Programmable Gate Array Devices-Based Secure Parallel Processing System for IoT Computer Vision Applications," in *2025 7th International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, Mohali, Punjab, India, 2025, pp. 1-8, doi: 10.1109/ISAECT68904.2025.11318787.
- [44] S. A. Khajeh, M. Saberikamarposhti, and A. M. Rahmani, "Real-time scheduling in IoT applications: A systematic review," *Sensors*, vol. 23, no. 1, p. 232, 2022, doi: 10.3390/s23010232.
- [45] S. Ahmed *et al.*, "Lightweight aes design for iot applications: Optimizations in fpga and asic with dfa countermeasure strategies," *IEEE Access*, vol. 13, pp. 22489-22509, 2025, doi: 10.1109/ACCESS.2025.3533611.
- [46] R. Parikh, "Low-Power Techniques for FPGA and ASIC Design: A Comprehensive Survey," *Preprints.org*, 2025, doi: 10.20944/preprints202503.1756.v1.
- [47] H. Lou, "Comparative analysis and enhancement of FPGA and ASIC performance in targeted applications," in *International Conference on Artificial Intelligence and Communication Technology*, 2024, pp. 295-308, doi: 10.1007/978-981-96-0096-0_23.
- [48] T. Soni, A. Kumar, I. Sharma, and M. K. Panda, "Design and implementation of low power multiplierless FIR filters on FPGA and ASIC," *International Journal of Electronics*, vol. 112, no. 9, pp. 1964-1986, 2025, doi: 10.1080/00207217.2024.2408797.

BIOGRAPHIES OF AUTHORS



Sababathy Palanisamy    is a researcher at the Cybersecurity Research Centre (CYRES), Universiti Sains Malaysia, Pulau Pinang, Malaysia. He specializes in cybersecurity, network security, and privacy-preserving technologies for modern communication systems. His research focuses on securing next-generation networks, including internet of things (IoT), fog and cloud computing, and distributed environments. He has contributed to numerous research projects addressing challenges such as secure authentication, data protection, and intrusion detection in large-scale and resource-constrained systems. His work often integrates advanced techniques such as machine learning, cryptographic protocols, and intelligent security frameworks to enhance system resilience and performance. He actively collaborates with international researchers and is committed to developing innovative and scalable cybersecurity solutions that meet the demands of emerging digital infrastructures. He can be contacted at email: sababathy@gmail.com.



Erum Ashraf    is a faculty member in the Faculty of Engineering Sciences and Technology at Bahria University, Islamabad, Pakistan. She is an active researcher in the fields of computer engineering and cybersecurity, with a strong focus on secure communication systems and intelligent network technologies. Her research interests include internet of things (IoT) security, fog and cloud computing, intrusion detection systems, and the application of machine learning techniques to enhance network performance and resilience. She has contributed to various academic publications addressing challenges in large-scale distributed systems, particularly in improving security, scalability, and efficiency. She is committed to advancing research and innovation through academic collaboration and practical implementation, aiming to develop robust and secure solutions for modern computing environments. She can be contacted at email: erum.buic@bahria.edu.pk.



Selvakumar Manickam    is currently working as an associate professor at National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia. His research interests include cybersecurity, internet of things, Industry 4.0, and machine learning. He has authored and co-authored more than 160 articles in journals, conference proceedings, and book reviews, and graduated 13 Ph.D.s. He has 10 years of industrial experience prior to joining academia. He is a member of technical forums at national and international levels. He also has experience building IoT, embedded, server, mobile, and web-based applications. He can be contacted at email: selva@usm.my.



Huda Mohammed Alsayednoor    is affiliated with the Department of Information Technology at Al-Arab University College, Basra, Iraq. She is an emerging researcher in the field of intelligent systems and cybersecurity, with a particular focus on secure and efficient computing in internet of things (IoT) and fog computing environments. She is actively engaged in collaborative research with international teams, aiming to develop practical and scalable solutions for next-generation computing environments. She is committed to advancing her expertise in intelligent network systems and contributing to high-impact research in the areas of fog computing and secure distributed architectures. She can be contacted at email: huda1994noor@gmail.com.



Dr. Mahmood A. Al-Shareeda    is a distinguished academic and researcher, currently serving as a lecturer at Southern Technical University in Basra, Iraq. He previously held a postdoctoral fellowship at the National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia (USM), in Penang, Malaysia. He completed his B.Sc., M.Sc., and Ph.D. degrees in Iraq, Lebanon, and Malaysia, respectively, reflecting a rich and diverse educational background across three nations. His outstanding contributions to research and academia have earned him global recognition, including being listed among the top 2% of scientists worldwide by Stanford University/Elsevier for the significant impact of his scholarly work. His expertise encompasses a wide range of research areas, including network monitoring, internet of things (IoT), vehicular ad-hoc network (VANET) security, and IPv6 security. He can be contacted at email: Mahmood.alshareedah@stu.edu.iq.



Shankar Karuppayah    is an Associate Professor, cybersecurity researcher, and academic at Universiti Sains Malaysia (USM), where he serves as a Senior Lecturer at the National Advanced IPv6 Centre (NAv6) and Deputy Director of the Cybersecurity Research Centre (CYRES). He obtained his Ph.D. in Cybersecurity from Technische Universität Darmstadt in 2016, with a dissertation focused on advanced monitoring of peer-to-peer (P2P) botnets. Prior to that, he completed his M.Sc. in Software Systems Engineering and B.Sc. (Hons) in Computer Science (Networking) from USM. His research expertise lies in cybersecurity, particularly in areas such as malware analysis, botnet monitoring, intrusion detection, internet of things (IoT) security, and distributed systems. He can be contacted at email: kshankar@usm.my.