Task Scheduling in Fog Computing-Powered Internet of Things Networks: A Review on Recent Techniques, Classification, and Upcoming Trends

Dongge TIAN
Department of Information Engineering, Hebei Chemical & Pharmaceutical College, Shijiazhuang 050000, China

***Abstract*—The Internet of Things (IoT) phenomenon influences daily activities by transforming physical equipment into smart objects. The IoT has achieved a wealth of technological innovations that were previously unimaginable. IoT application areas cover various sectors, including medical care, home automation, smart grids, and industrial operations. The massive growth of IoT applications causes network congestion because of the large volume of IoT tasks pushed to the cloud. Fog computing mitigates these transfers by placing resources near the edge. However, new challenges arise, such as limited computing power, high complexity, and the distributed characteristics of fog devices, negatively affecting the Quality of Service (QoS). Much research has been conducted to address these challenges in designing QoS-aware task scheduling optimization techniques. This paper comprehensively reviews task scheduling techniques in fog computing-powered IoT networks. We classify these techniques into heuristic-based, metaheuristic-based, and machine learning-based algorithms, evaluating their objectives, advantages, weaknesses, and performance metrics. Additionally, we highlight research gaps and propose actionable recommendations to address emerging challenges. Our findings offer a structured framework for researchers and practitioners to develop efficient, QoS-aware task scheduling solutions in fog computing environments.**

***Keywords*—Internet of Things; task scheduling; fog computing; quality of service; network congestion; optimization**

I. Introduction

The Internet of Things (IoT) phenomenon has changed the real world into a smart environment by turning everyday objects into smart objects/agents. This is accomplished by integrating sensors or microchips into these devices, along with internet connectivity [1]. These smart objects can independently interact and collect data, performing assigned duties [2]. The IoT is perceived as the upcoming model for ubiquitous computing and communication in the current technological environment. This ever-evolving environment is a network of billions of diverse, smart, connected devices that can revolutionize applications [3]. The applications of IoT range from individual home automation (smart homes) to overall city management (smart cities) [4]. It encompasses a variety of applications, including tracking high-precision agriculture and large-scale agricultural operations [5], monitoring individual building energy usage [6], and analyzing intelligent power grids [7].

The IoT also affects healthcare, providing personalized services for patients and the general public [8]. In addition, it also drives automation in industries and provides business information [9]. Moreover, IoT applications exist in weather forecasting and monitoring tools locally and remotely [10]. Cloud, fog, and edge technologies enable deploying distributed data processing solutions essential for the IoT paradigm [11].

Fog computing is a variation of cloud computing that distributes data across several geographical regions [12]. It positions the processing and communication resources closer to the network boundary, where several fog hubs are located. Proximity to end-users and IoT devices enhances performance and responsiveness [13]. Many applications are limited by cloud-centric architectures, particularly those demanding real-time performance within smart cities and buildings [14]. In such conditions, most data require processing, analysis, and storage on remote cloud servers. This dependency on remote resources may have detrimental effects on response time, privacy, elasticity, and system integrity [15].

The emergence of delay-sensitive and location-aware applications similarly reveals the weaknesses of cloud-based approaches, which often fail to meet their demands for high efficiency and low latency [16]. The presence of fog layers near IoT objects within a smart city environment decreases latency. This feature enables fog computing to meet demanding latency criteria effectively. Fog computing functions as a supplementary layer to the cloud, allowing advanced applications and services to be created and implemented [17].

Integrating IoT, fog, and cloud computing paradigms requires efficient task scheduling techniques. While these intricate and vast ecosystems emerge, it becomes essential to improve the entire tone of the environment, reduce delays, and efficiently utilize resources. Task scheduling solutions are required to control and allocate computational processes in clouds and other processing nodes, such as edges and IoT gadgets. As the demand for such environments increases, developing innovative and highly effective task scheduling initiatives to improve the performance and efficiency of IoT, fog, and cloud-based systems becomes crucial. To tackle this challenge, the current research adopts a multifaceted approach. The main contributions of this paper are as follows:

- We offer a detailed classification of task scheduling algorithms based on their impact on Quality of Service (QoS) for both users and fog service providers, providing a clear framework for understanding various approaches.

- Our study includes an extensive review of existing research, evaluating objectives, advantages, weaknesses, performance metrics, computing environments, and future work, thus providing a holistic view of the field.
- We identify current research gaps in the area of QoS-aware task scheduling in fog computing, offering a clear direction for future studies to address these gaps and advance the field.
- We provide explicit and actionable recommendations for future research based on identified trends and gaps, guiding scholars and practitioners in their efforts to develop more effective and efficient task scheduling algorithms.

Various performance metrics evaluate task-scheduling techniques, resulting in optimal resource utilization and efficient system performance. The most important factors are makespan, representing the total time taken to execute tasks; energy consumption, evaluating the power efficiency of the system; throughput, measuring how many tasks can be efficiently processed within a certain amount of time; reliability, guaranteeing no failure during the execution of a task; and latency, reflecting the reflection of delay time during the processing of tasks. These metrics are important in developing the effectiveness of task-scheduling approaches for IoT networks powered by fog computing and serve as a basis for our evaluation and classification in the study.

This investigation will reveal emerging research issues and future research prospects. To ensure this methodology remains coherent and specific, the following questions serve as a guideline for this research.

- What are the key performance metrics used to assess different task scheduling techniques?
- What are the recent trends and potential prospects in research on task scheduling for fog computing?
- What approaches and parameters can be used to tune fog computing task scheduling algorithms to achieve optimal resource utilization, throughput, and reliability without compromising energy consumption, makespan, and delay?

The remaining sections are organized in the following manner. Section II presents the basic insights on the effect of the IoT and a brief introduction to fog computing. Section III presents the classification of task scheduling techniques, including heuristic-based, metaheuristic-based, and machine learning-based algorithms. Section IV summarizes results from previous studies and their implications. Section V discusses potential future research for bridging identified gaps. Finally, Section VI concludes with actionable recommendations to further advance task scheduling for fog computing.

II. Background

A. *IoT and its Impact*

Numerous components, such as sensors, actuators, smartphones, and intelligent vehicles, are equipped with unique identities in IoT environments [18]. IoT optimizes daily activities by utilizing data to facilitate remote access control and configuration via cloud-based platforms. However, the increasing number of IoT devices poses a challenge: effective load balancing across these devices is essential to ensure optimal network performance [19]. This task is complicated primarily due to changing traffic patterns and the lack of a centralized network structure. Therefore, non-optimal load balancing is a major concern, which has motivated researchers to develop IoT load balancing and routing solutions.

The rapid proliferation of gadgets in modern networks and infrastructures has created a highly complex digital world. These systems create content having various packet sizes, inter-packet arrival intervals, and transmission lengths [20]. Therefore, managing and controlling traffic have emerged as significant concerns in many areas, such as healthcare, data centers, big data, smart cities, and other fields. Different communication protocols have been adopted to accommodate these diverse networks. Nevertheless, the lack of defined data formats and protocols poses a major obstacle to traffic control in traditional IoT architectures. The absence of consistency in communication protocols used by different IoT devices impedes the comprehensive analysis of data gathered from several sources [21].

The main responsibilities involve coordinating data transfer, ensuring it is timely, and optimizing network utilization while reducing bottlenecks, delays, and inefficiencies. Traffic management extends beyond particular settings and includes intelligent healthcare systems, urban environments, big data applications, and numerous other areas. Urban settings and smart healthcare utilize networked sensors and gadgets to acquire and collect relevant information. Therefore, it is necessary to deploy smart traffic control strategies to satisfy QoS criteria for these diverse technologies. To effectively control traffic flows in healthcare and automated transportation systems while also considering environmental sustainability, it is necessary to build complex algorithms and real-time data analytics to handle mobile IoT devices.

The IoT grows through the interconnectedness of numerous devices and sensors, resulting in ever-increasing data communication. The large amount of data might cause network congestion, especially in wireless networks with a natural tendency to encounter transmission obstacles [22]. Congestion in an IoT environment arises when the quantities of transmitted data are larger than the capacities of available transmission resources. This phenomenon has significant implications for load balancing routing protocols. There are two main forms of congestion: link-level and node-level.

At the link level, congestion can be defined as the arrival rate of the packets is more than the rate at which the packets are served, resulting in buffer overflow. This scenario is similar to one in which the amount of water pouring in exceeds the drainage system capacity [23]. On the other hand, node-level congestion happens when many active sensor nodes transmit packets simultaneously on the same channel, causing interference and preventing successful transmission [24].

The limited energy resources of devices present a substantial obstacle to implementing IoT networks. Energy-conscious routing protocols must confront complex issues associated with energy usage during sensing, data transmission, and receiving [25]. Data aggregation techniques can help reduce transmissions, but establishing the best level of aggregation is a complex operation that requires efficient solutions to maximize network device lifespan [26]. The natural diversity of IoT devices causes energy usage problems. These gadgets demonstrate notable disparities in computing power, energy profiles, and connectivity capabilities. Integrating these varied attributes into routing decisions is a considerable obstacle.

B. *Introduction to Fog Computing*

IoT deployment benefits from cloud computing in terms of computation, storage resources, and QoS constraints. It involves moving data to other servers at data centers, which is processed and returned to the end user [27]. Cloud computing also provides centralized virtual servers for data processing, storage, and analysis. These services are available on demand and are categorized as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

Nevertheless, IoT and the cloud components exchange more tasks and data, affecting the overall response time and resulting in network latency. The substantial physical separation between cloud-based IoT devices causes security problems. These delays can have severe consequences, especially in extremely sensitive tasks, such as real-time health monitoring, posing a danger to patients. Therefore, architectural plans are shifting from centralized data centers to distributed computing devices at the edge of the network. The decentralized nature of this mechanism also aims to eliminate the mentioned delays and security concerns.

Researchers have recently focused on investigating fog computing, a novel paradigm bridging the gap between IoT platforms and cloud computing. Fog computing leverages a decentralized architecture to reduce task transmission delays while upholding QoS requirements. Thus, this approach has developed into a sensible strategy for time-constrained operations within the IoT context. Unlike cloud computing, which relies on centralized servers to perform computations and relay outcomes to IoT devices, fog computing distributes these processes to fog nodes close to IoT devices to reduce latencies and enable increased response times in IoT applications.

The proximity of fog nodes to IoT devices provides further advantages, such as reduced total delay and increased protection of transmitted information. However, the resource limitations of fog devices necessitate offloading computationally intensive workloads to the cloud. Fig. 1 illustrates the overall structure, depicting the placement of IoT and edge devices with varying computational capabilities, cloud computing, and fog layers.

The goal of fog computing is to bring storage, transmission, and computing services closer to the network's edge. The proximity of the data center facilitates efficient data processing, low delays, and minimal bandwidth consumption. Fog devices, conversely, are characterized by lower processing, storage, and bandwidth capabilities due to their compact size. It is, therefore, important that all these constraints are considered when scheduling tasks to have an effective scheduling strategy. This problem is defined as non-deterministic polynomial-time hard (NP-hard) due to the optimization of the problem with many parameters, including energy consumption, task deadlines, cost, makespan, and response time. This designation indicates that finding the optimal solution becomes computationally intractable as the problem grows.

Certain constraints are crucial for both end-users and system designers. For instance, task deadlines represent a critical QoS parameter for end-users, while energy usage of fog nodes is a QoS requirement of concern for the fog service provider. Consequently, assigning IoT tasks to fog nodes and the cloud necessitates careful consideration of these diverse constraints.

III. Classification of Task Scheduling Techniques

This research adopts a systematic review approach to analyzing and classifying various task scheduling techniques in IoT networks powered by fog computing. Related literature concerning task scheduling and QoS improvements was searched from reputable databases such as IEEE Xplore, Springer, and Elsevier. Techniques identified were further classified into heuristic-based, metaheuristic-based, and machine learning-based approaches and assessed based on the key performance metrics of latency, energy consumption, throughput, and reliability. Critical reviews have been done regarding the objectives, advantages, and limitations of each technique. Research gaps were identified, with recommendations for improvement also given for actionable purposes.

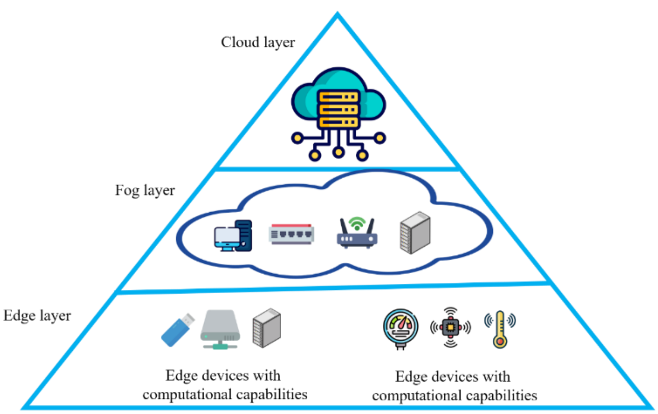

Fig. 1. IoT, edge, fog computing, and cloud computing architecture.

A. *Heuristic-based Algorithms*

As summarized in Table I, heuristic-based algorithms offer approximate solutions for computationally complex task scheduling problems within fog computing environments. These algorithms are specifically designed to efficiently manage the dynamic and heterogeneous characteristics inherent to fog environments.

Krivic, et al. [28] established the classification of IoT services, a crucial factor that directly influences scheduling algorithms. In addition, they introduced an innovative scheduling method that considers service context, user context, and processing devices. This approach allows for the efficient calculation of the most efficient schedule for executing service components in a distributed fog-to-cloud context. The effectiveness of the suggested algorithm was confirmed by simulations, in which its distinguishing innovation and dynamic scheduling were particularly highlighted.

Aburukba, et al. [29] formulated scheduling IoT service queries as an optimization problem to shorten the total service request latency. They employed integer programming to model the problem; however, due to its NP-hard nature, this approach becomes impractical for large-scale scenarios. To solve this problem, they combined an individualized version of the Genetic Algorithm (GA) as an efficient heuristic for scheduling IoT tasks, considering holistic latency optimization. They evaluated the performance of the designed GA using an evolutionary simulation model, which reflects the inherent dynamism of real IoT environments.

Ibrahim, et al. [30] introduce a load-balanced and delay-aware scheduling model for fog computing environments, particularly for critical IoT applications. The developed mechanism prioritizes minimizing task execution delays and maximizing task acceptance rates. Furthermore, the mechanism is designed to output an optimal outcome by ensuring uniform and minimal load imbalances to fog resources to improve resource utilization and lower the average response time.

Wireless Sensor Networks (WSNs) generate many tasks with varying priorities and durations in healthcare monitoring, transmitting them concurrently to fog computing platforms. This necessitates the implementation of an effective task scheduling algorithm capable of accurately prioritizing tasks according to their priority, regardless of their duration. Aladwani [31] introduced the Tasks Classification and Virtual Machines Categorization (TCVC) approach to improve the performance of static task scheduling algorithms. It classifies tasks by importance to patient health. The new method divides incoming IoT tasks into three levels of importance: high, average, and low.

Effective resource management is paramount for achieving optimal system performance, particularly concerning latency, within fog-cloud computing environments. Resource planning in such environments presents a computationally complex problem, classified as NP-hard. Khezri, et al. [32] investigated the optimization challenges associated with scheduling data-intensive jobs within fog-cloud based IoT systems, specifically focusing on maximizing job longevity. The proposed method starts by formulating the problem into an Integer Linear Programming (ILP) optimization scheme. Subsequently, a heuristic algorithm, Data-Locality Aware Job Scheduling in Fog-Cloud (DLJSF), is designed. Performance evaluations demonstrated that the proposed DLJSF algorithm achieves results closely approximating those obtained through the ILP model, with an average deviation of only 13.

TABLE I. SUMMARY OF HEURISTIC-BASED ALGORITHMS

Reference	Approach	Advantage	Disadvantage
[28]	Dynamic scheduling algorithm considering processing devices, user context, and service context	Significant increase in performance efficiency; adaptability to time-varying network conditions and QoS parameter changes	Complexity in implementation due to the need for constant monitoring and adjustments to QoS parameters
[29]	Individualized genetic algorithm for minimizing overall service request latency	Reduced overall latency	The genetic algorithm might be computationally expensive for large-scale scenarios
[30]	Load balancing mechanism prioritizing task execution delays and task acceptance rate	Improved resource utilization and lower average response time	Potential overhead in managing load balancing and ensuring task acceptance rates
[31]	Tasks classification and virtual machines categorization using the MAX-MIN scheduling algorithm	Improved performance in algorithm complexity, resource availability, execution time, waiting time, and finish time	The static nature of the approach might not handle highly dynamic environments effectively
[32]	Data-locality aware task scheduling in fog-cloud derived from an ILP optimization model	Results closely approximate the ILP model with a 13% average deviation; outperforms local processing by 99.16%	The ILP-based model might be complex, computationally intensive, and requires effective data locality awareness.
[33]	Priority-Aware Semi-Greedy (PSG) and PSG with Multistart (PSG-M) procedures	High performance on makespan, deadline violation time, energy consumption, and deadline compliance	Balancing multiple objectives (energy usage and QoS) can be challenging and may require fine-tuning for different scenarios
[34]	Heuristic for dynamic resource scheduling and allocation of real-time IoT workflows	Superior performance compared to static provisioning; real-time data awareness	Complexity in implementing dynamic resource provisioning and maintaining real-time data awareness.

Azizi, et al. [33] explored the issue of task scheduling in fog computing to find a compromise between reducing energy consumption in fog points and maintaining the QoS standards for IoT operations. They mathematically model the problem of optimizing these conflicting criteria as a multi-objective optimization problem. They also focused on minimizing deadline violation time in their approach, which they handled by proposing two new semi-greedy based algorithms: Priority-Aware Semi-Greedy (PSG) algorithm and a PSG with Multistart (PSG-M) procedure.

Stavrinides and Karatza [34] proposed a dynamic resource provisioning mechanism for cloud resources within a three-layer IoT-fog-cloud framework. This approach prioritizes real-time data awareness and dynamic scaling to optimize resource allocation. Additionally, they introduced a heuristic for scheduling instant IoT tasks. The efficacy of the suggested scheme was measured through experiments that compared its performance against a static provisioning strategy. These simulations employed various workload patterns to assess the impact on the framework's performance under different provisioning scenarios.

B. Metaheuristic-based Algorithms

Building upon heuristic approaches, metaheuristic-based algorithms leverage advanced strategies to efficiently explore and exploit the search space. These algorithms strive to identify near-optimal solutions while exhibiting improved convergence rates, as shown in Table II.

Abdel-Basset, et al. [35] introduced an energy-conscious task scheduling method for fog environments based on the Harris Hawks Optimization (HHO) algorithm integrated with Local Search, called HHOLS. HHOLS optimizes the QoS in IoT applications by focusing on energy efficiency. Their work commences with a detailed description of the highly virtualized layered fog computing model, emphasizing its heterogeneous architectural characteristics. To address the non-linear character of the task scheduling problem, they incorporated a scaling and normalization stage to adapt the standard Harris Hawks optimization algorithm.

Service execution plays a vital role in IoT networks, which is also an important problem in scheduling services in fog computing. Consequently, fog infrastructure provides the execution environment for devices with limited computational capability. A fog environment comprises many fog nodes that can be some-edge servers, cloudlets, small-size ISPs, and caching nodes offering user-requested services. Najafizadeh, et al. [36] offered a privacy-preserving task scheduling architecture for IoT systems based on service execution to overcome these problems. In this design, a multi-objective algorithm is proposed to lower both service cost and execution time simultaneously.

Abd Elaziz, et al. [37] suggested AEOSSA, an alternative task scheduling approach for managing IoT tasks within a cloud-fog computing context. This approach builds upon a modified Artificial Ecosystem-Based Optimization (AEO) algorithm. The modified algorithm incorporates operators derived from the Salp Swarm Algorithm (SSA) to augment the exploitation capabilities of AEO in the search for optimum results for the task scheduling problem. An evaluation of AEOSSA is performed on some synthetic and real datasets that include a variety of computation sizes.

TABLE II. SUMMARY OF METAHEURISTIC-BASED ALGORITHMS

Reference	Approach	Advantage	Disadvantage
[35]	Harris hawks optimization with local search	Optimizes QoS in IoT applications focusing on energy efficiency	Complexity due to normalization, scaling, and local search phases
[36]	Multi-objective algorithm for privacy-preserving task scheduling	Minimizes service execution time and cost; maintains the privacy of IoT devices; performs well across different service composition complexities	Higher computational overhead due to multi-objective optimization
[37]	Modified artificial ecosystem-based optimization with salp swarm algorithm	Superior performance in makespan and throughput; effective for synthetic and real datasets	Potentially higher resource consumption due to extensive exploitation capabilities
[38]	Energy-aware model with arithmetic optimization algorithm and marine predators algorithm	Significant reductions in energy consumption and makespan	Increased complexity and potential for higher computational cost
[39]	Multi-cloud to multi-fog architecture with dynamic threshold strategy	Decreases service latency and increases fog node efficiency; achieves energy balance	Complexity in implementation and real-time dynamic scheduling
[40]	CHMPAD algorithm combining marine predators algorithm and disruption operator	Prevents local optimization; improves exploitation properties; significant reductions in makespan and throughput	Increased complexity and resource demands
[41]	Two-tiered approach with PSO and particle swarm genetic joint optimization artificial bee colony	Optimal load balancing and task scheduling; lower delay and energy consumption	Higher computational complexity due to multi-tiered strategy
[42]	Multi-tiered scheduling framework with Naïve Bayes classifier	Effective task classification and placement; enhances QoS parameters	Requires precise training data for classifier accuracy
[43]	Directed non-dominated sorting genetic algorithm	Minimizes energy consumption and response times; balances exploration and exploitation	Potential for higher computational overhead
[44]	Hunger games search with marine predators algorithm	Reduces energy consumption and makespan; effective for various workload traces	Complexity in algorithm integration and evaluation
[45]	Multi-objective gravitational search algorithm with star-quake operator	Reduces makespan, energy consumption, and cost; prevents local optima	Increased computational complexity and resource demands
[46]	Various algorithms, including machine learning and nature-inspired metaheuristics	Optimizes resource allocation, minimizes energy consumption and latency, and meets deadlines; consistent performance improvements	Varied complexity depending on the specific algorithm used

Abd Elaziz, et al. [38] overcame the task scheduling issue in fog computing by proposing an energy-aware model using a variant of the Arithmetic Optimization Algorithm (AOA) known as AOAM. Optimizing the makespan metric, they ensured that user QoS was the top priority. The authors integrated search operators inspired by the Marine Predators Algorithm (MPA) to overcome the limitations of the traditional AOA. This modification encourages a wider range of solutions and avoids becoming stuck in suboptimal solutions. The efficacy of the proposed AOAM was validated through simulations that employed various parameters.

Luo, et al. [39] have suggested a unique multi-cloud to multi-fog model that involves two service models with containerization technology, aiming to optimize fog resource usage and control service latency. On the other hand, the task scheduling algorithm provided in their proposal is particularly suitable for achieving an energy balance. Additionally, the algorithm takes the terminal device transmission energy requirements into account. It applies a flexible threshold algorithm to achieve real-time request scheduling while ensuring an energy balance state of the terminal device, thereby effectively avoiding transmission delays.

Attiya, et al. [40] presented a novel fog computing application-aware task scheduling algorithm called CHMPAD. It overcomes existing issues with the Chimp Optimization Algorithm by combining two key components of different algorithms: the Marine Predators Algorithm and a disruption operator. CHMPAD aims to prevent local optimization and improve the exploitation properties of the base ChOA algorithm. The applicability and effectiveness of CHMPAD are evaluated through extensive experiments performed on synthetic and real-world workloads.

Liu, et al. [41] developed a novel resource scheduling strategy for fog computing environments. This two-tiered approach optimizes load balancing and task scheduling to decrease energy consumption and execution time. The first tier leverages the Particle Swarm Optimization (PSO) algorithm for balancing loads within a fog cluster. This optimization seeks to identify the ideal distribution of tasks across fog nodes, minimizing computation time and energy usage. Building upon this foundation, the authors propose a novel Particle Swarm Genetic Joint Optimization Artificial Bee Colony (PGABC) algorithm. PGABC tackles the challenge of task scheduling across multiple fog clusters, utilizing the time and energy consumption data obtained from the initial load balancing phase.

Kaur, et al. [42] proposed a multi-tiered scheduling framework for managing IoT application tasks. This framework prioritizes QoS parameters to achieve optimal task placement. The framework operates on two levels: fog environment and fog node selection. The specific fog environment in which the task will be executed is set at the first level. Several factors, such as availability, physical distance, latency, and throughput, are used to choose an environment. After choosing the fog environment, a particular fog node is selected for analysis. They implemented a Naïve Bayes classifier to classify the task category (Compute-intensive, Memory-intensive, or GPU-intensive) based on the probability triad (C, M, G).

Mousavi, et al. [43] formulated a constrained bi-objective optimization problem for task scheduling in fog computing environments. This formulation aims to achieve two critical goals simultaneously: minimizing server energy consumption and reducing overall response times. To address this challenge, the authors proposed a novel Directed Non-dominated Sorting Genetic Algorithm (D-NSGA-II). This algorithm builds upon the foundation of NSGA-II, a well-established multi-objective optimization technique. The key innovation lies in the introduction of a new recombination operator. This operator empowers D-NSGA-II to regulate the selection pressure exerted on candidate solutions, thereby striking a balance between the algorithm's exploration and exploitation capabilities.

Attiya, et al. [44] proposed a novel task scheduling algorithm, HGSMPA, specifically designed for cloud-fog computing environments within the IoT domain. Their approach leverages the Hunger Games Search (HGS) algorithm as a foundation. The authors incorporated elements from the MPA to enhance the exploitation capabilities inherent in HGS. The efficacy of HGSMPA was validated through experimental evaluations that employed various workload traces, both synthetic and real-world. The results convincingly demonstrate the superiority of HGSMPA compared to existing scheduling algorithms.

Ahmadabadi, et al. [45] introduced a novel multi-objective task scheduling approach for fog-cloud computing systems. Their approach addresses three critical objectives simultaneously: minimizing monetary cost, energy consumption, and makespan. To achieve these goals, the authors proposed a new multi-objective function that incorporates all three objectives. Furthermore, they introduced a novel operator called star-quake, specifically designed for the Multi-Objective Gravitational Search Algorithm (MOGSA). This operator balances the algorithm's capabilities, such as selection pressure, exploration, and exploitation.

Alsamarai, et al. [46] have significantly improved task scheduling in fog-cloud computing environments for IoT applications. Their proposed algorithms address various challenges, including optimizing resource allocation (e.g., CHMPAD, DLJSF), minimizing energy consumption and latency (e.g., PGABC, Quality-aware Energy Efficient Scheduling), and meeting task deadlines (e.g., Bandwidth-Deadline Algorithm). They leverage a variety of techniques, including machine learning (PSO, ANN), heuristic approaches (genetic algorithms), and nature-inspired metaheuristics (Gravitational Search Algorithm, Ant Colony Optimization) to achieve these improvements.

C. *Machine Learning-based Algorithms*

Machine learning-based algorithms exploit historical data and learning models to predict near-optimal scheduling decisions [47]. Techniques such as reinforcement learning and neural networks have demonstrated significant potential in adapting to the dynamic characteristics of fog computing environments [48], as shown in Table III.

Bhatia, et al. [49] proposed a novel quantized approach for scheduling heterogeneous tasks within fog computing applications. The approach is built on a node-specific metric, the Node Computing Index (NCI), used to measure individual fog nodes' computational capability. They also proposed a QCI-Neural Network Model that forecasts the best available fog node for real-time execution of heterogeneous tasks. To validate the proposed approach, the authors conducted simulations in different scenarios.

Ali, et al. [50] tackled enhancing the overall efficiency of executing tasks for IoT applications. Their methodology revolves around selecting real-time jobs well-suited for execution at the fog layer. A fuzzy logic-based task scheduling algorithm is modeled for a fog-cloud computing environment.

This algorithm offers a smart scheme of allocating submitted tasks to the processing units within the fog layer. Heterogeneous resources can be found in fog.

Lim [51] addressed the low latency task execution in small-scale fog computing deployments. Their approach is based on a novel task scheduling strategy using partitioned Artificial Neural Networks (ANNs). Such partition allows parallel learning and hyperparameter optimization across different edge servers. This parallelism significantly reduces scheduling times and contributes to achieving desired service level objectives.

Aburukba, et al. [52] proposed a task scheduling solution for a three-tier fog computing architecture. This approach prioritizes maximizing the number of requests that meet their deadline requirements. To achieve this goal, the authors introduce an optimization model formulated using Mixed Integer Programming (MIP). This model aims to minimize the number of missed deadlines. The efficacy of the model was validated using an exact solution technique. However, the authors acknowledge that the task scheduling problem is NP-hard, rendering exact solutions impractical for typical fog computing environments due to problem size.

TABLE III. SUMMARY OF MACHINE LEARNING-BASED ALGORITHMS

Reference	Approach	Advantage	Disadvantage
[49]	Node computing index and QCI-neural network model	Significantly improved performance in execution delay, sensitivity, and precision; suitable for heterogeneous tasks	High computational complexity may require substantial training data.
[50]	Fuzzy logic-based task scheduling algorithm	Outperformed existing algorithms in task success ratio, makespan, average turnaround time, and delay rate; efficient for heterogeneous resources	Potential complexity in defining fuzzy rules may not scale well with large task sets.
[51]	Partitioned artificial neural network	Reduced scheduling times, maintained low energy consumption, achieved desired service level objectives	Limited scalability to larger fog computing environments, potential overhead in partitioning.
[52]	Mixed Integer Programming and genetic algorithm	Significant reduction in missed deadlines, effective for NP-hard scheduling problems; superior performance compared to round-robin and priority scheduling	The exact solution technique is impractical for large problem sizes, and the heuristic approach may not always find the global optimum.
[53]	Statistical techniques (moving averages, Heikin-Ashi patterns)	Enhanced precision in scheduling times, optimized task allocation across edge and fog nodes	The applicability of financial patterns to computing tasks may not be universally effective, and there is potential for increased computational overhead.
[54]	K-Means clustering and fuzzy logic	Accurate identification of groups, adapts to changing task distributions, improved execution time, response time, and network usage	Complexity in implementation and potential high computational overhead in large-scale dynamic environments
[55]	Distributed deep reinforcement learning with asynchronous proximal policy optimization	Fast convergence rate, high flexibility, upgradeability, and better time complexity in execution	Greedy nature of existing techniques and complexity in managing distributed experience trajectories
[56]	A2C-DRL based real-time task scheduling for edge-cloud environments	Simultaneous learning at multiple servers, flexibility with assignable hyperparameters, and superior load balancing	Complexity in defining reward functions and update policies
[57]	DRL-based algorithm for scheduling IoT applications	Adaptive response time, load balancing, significant cost reduction in execution and load balancing	Initial training phase might be resource-intensive, complexity in implementation on different platforms

The rapid rise in bandwidth requirements and computational load of the IoT has created opportunities for fog computing. However, maintaining the QoS of the data transfer process at an efficient cost in fog-based IoT networks remains a significant challenge. Potu, et al. [53] propose a novel scheduling algorithm that optimizes task allocation across edge and fog nodes. The proposed model integrates various statistical techniques, including moving averages and Heikin-Ashi patterns frequently employed in financial markets to visualize trends.

The basic scheduling strategies designed for specials global cloud model do not really cope with static nature, heterogeneity, and resource constraints of the fog nodes. Sheikh, et al. [54] have addressed these limitations through the development of a new machine learning based approach that is aimed at dynamically allocating tasks in respect of the evolving status in the fog environment. Their approach builds on basics of K-Means clustering algorithm substantiated by fuzzy logic, which can be considered an example of an unsupervised learning. Overall, this approach economically categories fog nodes based on resource and workload distribution. The proposed method builds on the strong points of the K-Means clustering that provides accurate identification of groups and fuzzy logic that allows one to adapt to changes concerning the distribution of tasks among the fog nodes.

Deep Reinforcement Learning (DRL) has recently gained traction in addressing complex service offloading problems. However, existing techniques are greedy in nature and are primarily designed for centralized problem formulation which results in slow convergence towards the global solution. In addition, data dependencies that are preconceived and QoS requirements inherent within the service components do not facilitate offloading. To overcome such limitations, Goudarzi, et al. [55] developed a distributed DRL strategy formulated through an actor-critic architecture named Asynchronous Proximal Policy Optimization (APPO). Thereby, it contributes to creating a multitude of possible distributed experience trajectories to take place. Moreover, the authors use off-policy correction methods we have reviewed include PPO clipping and V-trape so as to enhance the rate of convergence to the optimal service offloading solutions.

Resource management in mobile edge and cloud systems often presents complex online decision-making challenges. Effective solutions necessitate real-time understanding of both workload and environment to facilitate the efficient utilization of distributed resources. However, geographically dispersed resources, limited capacity, unpredictable task characteristics, and network hierarchy inherent to edge environments significantly hinder efficient job scheduling. The above dynamic scenarios make heuristic-based methods inadequate since they are not easy to generalize or modify as would be required at times. One such unutilized yet potentially very beneficial technique is the DRL that names Advantage Actor-Critic (A2C). A2C learns quickly in environments with little data while DRL gains its knowledge from situations within the environment and applies them to make a decision. To address these challenges, Lu, et al. [56] propose an A2C-DRL based real-time task scheduling technique specifically designed for stochastic edge-cloud environments.

Wang, et al. [57] introduced a Deep Reinforcement Learning (DRL)-based algorithm for scheduling IoT applications, termed DRLIS. This approach is targeted to provide adaptive and efficient response time for wide range of IoT applications as well as the load balancing among the edge/fog servers. The authors incorporated DRLIS as an operational scheduler in the FogBus2, which is a function-as-a-service platform in the development of moving from edge to fog to cloud serverless computing model. The results of varied experiences indicate that the DRLIS bears a higher impact on the improvement of the execution cost of IoT applications.

IV. Results and Discussion

A review and analysis of different task-scheduling techniques reveal tremendous variability in their performance based on the underlying methodologies and QoS metrics they try to optimize. Due to their simplicity, heuristic-based algorithms utilize low computational overhead; thus, they can be applied to only small-scale and resource-limited environments. Most of these techniques fail to optimize multiple QoS parameters like latency, energy consumption, and throughput simultaneously; hence, their usage is quite impractical in dynamic and large-scale fog computing scenarios. Contrarily, metaheuristic-based algorithms demonstrate much stronger adaptability in finding near-optimal solutions for complex scheduling problems. Despite the better performance, these algorithms usually introduce higher computation overhead, which may not be affordable for real-time applications.

While advanced algorithms based on machine learning leverage predictive and adaptive capabilities to optimize task scheduling dynamically, techniques such as reinforcement learning and deep neural networks have been promising in achieving significant reductions in latency and energy consumption while maintaining high throughput. For example, reinforcement learning-based models can predict the pattern of task arrivals and resource availability to enable proactive scheduling. However, most machine learning techniques are implemented in a fog computing environment with extensive training in complex data computation resources and feature engineering, challenging widespread adaptation. Besides, machine learning interpretability may be one of the barriers if transparency in decision-making is essential.

Comparative analyses reveal that no method covers the fog computing-powered IoT network to date for all the challenges combined. Instead, heuristic and metaheuristic algorithms are more appropriate for scenarios with specific resource constraints, while machine learning-based approaches best apply to dynamic and complex environments. The hybridization of such techniques, though in very few instances, points out a bright future direction that can leverage the simplicity and efficiency of the heuristic approach together with adaptability and optimization capabilities from the metaheuristic and machine-learning-based techniques. Furthermore, much emphasis on standard frameworks is required to evaluate the benchmark techniques with regard to task scheduling properly; this ensures coherence in the performances reported through different scenarios. These insights emphasize the vital need for further research that should result in novel hybrid techniques applicable to the new demands put forward by IoT systems driven by fog computing.

Through extensive analysis of various methods for task scheduling in fog-cloud computing environments for IoT applications, several research gaps and limitations in prior studies have been identified. These limitations can include high run times, failure to meet the study's objectives, or negative impacts on other performance metrics. Common shortcomings include missing details on simulation parameters, comparisons with outdated algorithms, using small datasets for evaluation, omitting definitions of evaluation parameters and equations, neglecting relevant evaluation factors, and lacking results to support performance claims.

A critical limitation identified is using outdated benchmark algorithms for comparison in some studies. This makes it difficult to assess the efficiency of the proposed methods definitively. Additionally, several studies employed small datasets (fewer than 100 tasks) or omitted data set size information entirely. This raises concerns about the proposed algorithms' ability to handle real-world workloads with high throughput.

Our analysis also revealed a focus on specific objectives and performance metrics. Energy consumption emerged as the primary objective in many studies, followed by minimizing makespan, delay, and cost. Conversely, response time, resource utilization, deadline violation, security, and reliability received less attention. This focus is reflected in the most studied metrics: makespan, energy consumption, and cost. Most of the investigated algorithms were multi-objective, focusing on optimizing combinations like makespan and cost, makespan and energy, or delay and energy simultaneously.

V. Future Directions

As fog computing continues to evolve, several emerging trends and research directions are shaping the landscape of task scheduling algorithms. These advancements aim to enhance the efficiency, scalability, and adaptability of fog computing systems to meet the growing demands of IoT applications. The future of task scheduling in fog computing is poised to leverage more sophisticated machine learning and Artificial Intelligence (AI) techniques. Reinforcement learning, deep learning, and federated learning are expected to play a significant role in developing more adaptive and intelligent scheduling algorithms. These approaches can enable real-time learning and decision-making, improving the allocation of resources and the overall performance of fog environments.

The collaboration between edge and cloud resources is anticipated to become more seamless, providing a hybrid model that optimally distributes tasks based on their computational and latency requirements. Future research will focus on developing algorithms that dynamically balance the load between edge and cloud, considering network conditions, energy consumption, and application-specific constraints. Energy efficiency will remain a critical concern in fog computing, particularly with the increasing number of connected devices and data-intensive applications. The research will continue to explore energy-aware scheduling algorithms that minimize power consumption without compromising performance. Sustainable computing practices, such as using renewable energy sources and energy-harvesting techniques, will also gain more attention.

With the proliferation of IoT devices and the sensitivity of the data they generate, ensuring security and privacy in task scheduling is paramount. Future research will delve into developing secure scheduling algorithms that incorporate encryption, anonymization, and other privacy-preserving techniques. These solutions must safeguard data integrity and confidentiality while maintaining efficient resource utilization. Integrating real-time analytics and predictive modeling into task scheduling algorithms will enhance responsiveness and accuracy. Using historical data and real-time monitoring, these algorithms can predict workload patterns, detect anomalies, and proactively adjust resource allocation, improving system reliability and performance.

Future scheduling algorithms will increasingly adopt multi-objective optimization techniques to balance various conflicting performance metrics, such as latency, throughput, energy consumption, and cost. Research will focus on developing algorithms that can effectively navigate the trade-offs between these objectives, providing optimal solutions that meet diverse application requirements. As quantum computing technologies mature, their integration into fog computing task scheduling could revolutionize the field. Quantum algorithms have the potential to solve complex optimization problems more efficiently than classical algorithms, offering unprecedented improvements in scheduling performance and resource utilization.

Developing standardized frameworks and protocols for task scheduling in fog computing ensures interoperability between different devices and platforms. Future research will explore ways to create universally accepted standards that facilitate seamless integration and collaboration across heterogeneous fog and edge environments. User-centric and context-aware scheduling algorithms that consider the specific needs and preferences of end-users will become more prevalent. These algorithms will consider contextual information, such as user location, device capabilities, and application-specific requirements, to deliver personalized and efficient task scheduling solutions.

Federated learning, a decentralized machine learning approach where model training occurs locally on edge devices, will gain prominence in fog computing environments. The research will explore federated learning techniques for collaborative model training across distributed edge nodes, enabling privacy-preserving and resource-efficient machine learning. These approaches will empower edge devices to perform predictive analytics and decision-making tasks autonomously without relying heavily on centralized cloud servers. The integration of blockchain technology into task scheduling algorithms will enhance security, transparency, and trust in fog computing environments. Future research will investigate blockchain-based scheduling mechanisms that ensure verifiable task execution, prevent tampering or manipulation of scheduling decisions, and enable secure peer-to-peer transactions between edge devices. These blockchain-enabled solutions will facilitate decentralized task allocation and resource sharing while preserving data integrity and privacy.

VI. Conclusion

IoT technology applies to promote immense interconnectedness to data-driven functions like smart homes, cities, industrial automation, and health services. At the time of growth, this explosive creation of data created newer challenges for traditional models in cloud computing: latencies at high traffic volume and constricts scalability. Emerging as complementary mechanisms to remedy several limitations brought upon by traditional clouds in view of the huge impact created by IoT sensors continuously generating huge loads of information that requires computation, sometimes really urgent, Fog Computing extends the computational processes further toward network boundaries.

The present research gave an extensive review of different techniques for task scheduling in fog computing environments and broadly classified these techniques into three main categories, namely heuristic-based approaches, metaheuristic-based methods, and machine learning-based approaches. This research further analyzed the effects of different techniques on key QoS metrics related to latency, energy consumption, makespan, and reliability and, therefore, provided pragmatic insights into strengths, weaknesses, and applicability. The results revealed that these techniques can adapt dynamically to changing network conditions and workload demands, optimizing resource utilization and service quality in fog-enabled IoT systems.

Apart from indicating gaps, the study also identified several innovative solutions with regard to needs in proposals on hybrid techniques, along with standardized frameworks concerning the evaluation perspective. These lessons then provided concrete guidelines for both the researcher and the practitioner in creating algorithms on next-generation task scheduling at fog computing-powered IoT with efficiency and scalability, thus guaranteeing enhanced QoS toward paving the right path for R-IoT.

REFERENCES

[1] B. Pourghebleh, V. Hayyolalam, and A. A. Anvigh, "Service discovery in the Internet of Things: review of current trends and research challenges," Wireless Networks, vol. 26, no. 7, pp. 5371-5391, 2020.

[2] B. Pourghebleh and N. J. Navimipour, "Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research," Journal of Network and Computer Applications, vol. 97, pp. 23-34, 2017, doi: https://doi.org/10.1016/j.jnca.2017.08.006.

[3] B. Pourghebleh, K. Wakil, and N. J. Navimipour, "A comprehensive study on the trust management techniques in the Internet of Things," IEEE Internet of Things Journal, vol. 6, no. 6, pp. 9326-9337, 2019, doi: https://doi.org/10.1109/JIOT.2019.2933518.

[4] A. Ullah et al., "Smart cities: The role of Internet of Things and machine learning in realizing a data-centric smart environment," Complex & Intelligent Systems, vol. 10, no. 1, pp. 1607-1637, 2024.

[5] S. Atalla et al., "Iot-enabled precision agriculture: Developing an ecosystem for optimized crop management," Information, vol. 14, no. 4, p. 205, 2023.

[6] R. Selvaraj, V. M. Kuthadi, and S. Baskar, "Smart building energy management and monitoring system based on artificial intelligence in smart city," Sustainable Energy Technologies and Assessments, vol. 56, p. 103090, 2023.

[7] N. Renugadevi, S. Saravanan, and C. N. Sudha, "IoT based smart energy grid for sustainable cites," Materials Today: Proceedings, vol. 81, pp. 98-104, 2023.

[8] F. Kamalov, B. Pourghebleh, M. Gheisari, Y. Liu, and S. Moussa, "Internet of medical things privacy and security: Challenges, solutions, and future trends from a new perspective," Sustainability, vol. 15, no. 4, p. 3317, 2023, doi: https://doi.org/10.3390/su15043317.

[9] K. C. Rath, A. Khang, and D. Roy, "The Role of Internet of Things (IoT) Technology in Industry 4.0 Economy," in Advanced IoT Technologies and Applications in the Industry 4.0 Digital Economy: CRC Press, 2024, pp. 1-28.

[10] Ş. M. Kaya, B. İşler, A. M. Abu-Mahfouz, J. Rasheed, and A. AlShammari, "An intelligent anomaly detection approach for accurate and reliable weather forecasting at IoT edges: A case study," Sensors, vol. 23, no. 5, p. 2426, 2023.

[11] S. Ahmad, I. Shakeel, S. Mehfuz, and J. Ahmad, "Deep learning models for cloud, edge, fog, and IoT computing paradigms: Survey, recent advances, and future directions," Computer Science Review, vol. 49, p. 100568, 2023.

[12] K. Behravan, N. Farzaneh, M. Jahanshahi, and S. A. H. Seno, "A comprehensive survey on using fog computing in vehicular networks," Vehicular Communications, p. 100604, 2023.

[13] N. Keshari, D. Singh, and A. K. Maurya, "A survey on Vehicular Fog Computing: Current state-of-the-art and future directions," Vehicular Communications, vol. 38, p. 100512, 2022.

[14] V. Hayyolalam, B. Pourghebleh, A. A. P. Kazem, and A. Ghaffari, "Exploring the state-of-the-art service composition approaches in cloud manufacturing systems to enhance upcoming techniques," The International Journal of Advanced Manufacturing Technology, vol. 105, no. 1-4, pp. 471-498, 2019.

[15] V. Hayyolalam, B. Pourghebleh, and A. A. Pourhaji Kazem, "Trust management of services (TMoS): investigating the current mechanisms," Transactions on Emerging Telecommunications Technologies, vol. 31, no. 10, p. e4063, 2020.

[16] F. Yunlong and L. Jie, "Incentive approaches for cloud computing: challenges and solutions," Journal of Engineering and Applied Science, vol. 71, no. 1, p. 51, 2024.

[17] G. Javadzadeh and A. M. Rahmani, "Fog computing applications in smart cities: A systematic survey," Wireless Networks, vol. 26, no. 2, pp. 1433-1457, 2020.

[18] B. Pourghebleh and V. Hayyolalam, "A comprehensive and systematic review of the load balancing mechanisms in the Internet of Things," Cluster Computing, pp. 1-21, 2019.

[19] B. Pourghebleh, N. Hekmati, Z. Davoudnia, and M. Sadeghi, "A roadmap towards energy-efficient data fusion methods in the Internet of Things," Concurrency and Computation: Practice and Experience, p. e6959, 2022, doi: https://doi.org/10.1002/cpe.6959.

[20] A. A. Anvigh, Y. Khavan, and B. Pourghebleh, "Transforming Vehicular Networks: How 6G can Revolutionize Intelligent Transportation?," Science, Engineering and Technology, vol. 4, no. 1, 2024.

[21] B. A. Begum and S. V. Nandury, "Data aggregation protocols for WSN and IoT applications–A comprehensive survey," Journal of King Saud University-Computer and Information Sciences, vol. 35, no. 2, pp. 651-681, 2023.

[22] M. A. Rahim, M. A. Rahman, M. M. Rahman, A. T. Asyhari, M. Z. A. Bhuiyan, and D. Ramasamy, "Evolution of IoT-enabled connectivity and applications in automotive industry: A review," Vehicular Communications, vol. 27, p. 100285, 2021.

[23] H. Farag and Č. Stefanovič, "Congestion-aware routing in dynamic iot networks: A reinforcement learning approach," in 2021 IEEE Global Communications Conference (GLOBECOM), 2021: IEEE, pp. 1-6.

[24] S. Bansal and D. Kumar, "Distance-based congestion control mechanism for CoAP in IoT," IET Communications, vol. 14, no. 19, pp. 3512-3520, 2020.

[25] O. L. López et al., "Energy-sustainable iot connectivity: Vision, technological enablers, challenges, and future directions," IEEE Open Journal of the Communications Society, 2023.

[26] M. Zhang, Y. Li, Y. Ding, and B. Yang, "A Lightweight and Robust Multi-Dimensional Data Aggregation Scheme for IoT," IEEE Internet of Things Journal, 2023.

[27] V. Hayyolalam, B. Pourghebleh, M. R. Chehrehzad, and A. A. Pourhaji Kazem, "Single-objective service composition methods in cloud manufacturing systems: Recent techniques, classification, and future trends," Concurrency and Computation: Practice and Experience, vol. 34, no. 5, p. e6698, 2022.

[28] P. Krivic, M. Kusek, I. Cavrak, and P. Skocir, "Dynamic scheduling of contextually categorised internet of things services in fog computing environment," Sensors, vol. 22, no. 2, p. 465, 2022.

[29] R. O. Aburukba, M. AliKarrar, T. Landolsi, and K. El-Fakih, "Scheduling Internet of Things requests to minimize latency in hybrid Fog–Cloud computing," Future Generation Computer Systems, vol. 111, pp. 539-551, 2020.

[30] M. Ibrahim, Y. Lee, and D.-H. Kim, "DALBFog: Deadline-Aware and Load-Balanced Task Scheduling for the Internet of Things in Fog Computing," IEEE Systems, Man, and Cybernetics Magazine, vol. 10, no. 1, pp. 62-71, 2024.

[31] T. Aladwani, "Scheduling IoT healthcare tasks in fog computing based on their importance," Procedia Computer Science, vol. 163, pp. 560-569, 2019.

[32] E. Khezri, R. O. Yahya, H. Hassanzadeh, M. Mohaidat, S. Ahmadi, and M. Trik, "DLJSF: Data-Locality Aware Job Scheduling IoT tasks in fog-cloud computing environments," Results in Engineering, vol. 21, p. 101780, 2024.

[33] S. Azizi, M. Shojafar, J. Abawajy, and R. Buyya, "Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach," Journal of network and computer applications, vol. 201, p. 103333, 2022.

[34] G. L. Stavrinides and H. D. Karatza, "Scheduling real-time IoT workflows in a fog computing environment utilizing cloud resources with data-aware elasticity," in 2021 Sixth International Conference on Fog and Mobile Edge Computing (FMEC), 2021: IEEE, pp. 1-8.

[35] M. Abdel-Basset, D. El-Shahat, M. Elhoseny, and H. Song, "Energy-aware metaheuristic algorithm for industrial-Internet-of-Things task scheduling problems in fog computing applications," IEEE Internet of Things Journal, vol. 8, no. 16, pp. 12638-12649, 2020.

[36] A. Najafizadeh, A. Salajegheh, A. M. Rahmani, and A. Sahafi, "Privacy-preserving for the internet of things in multi-objective task scheduling in cloud-fog computing using goal programming approach," Peer-to-Peer Networking and Applications, vol. 14, pp. 3865-3890, 2021.

[37] M. Abd Elaziz, L. Abualigah, and I. Attiya, "Advanced optimization technique for scheduling IoT tasks in cloud-fog computing environments," Future Generation Computer Systems, vol. 124, pp. 142-154, 2021.

[38] M. Abd Elaziz, L. Abualigah, R. A. Ibrahim, and I. Attiya, "IoT workflow scheduling using intelligent arithmetic optimization algorithm in fog computing," Computational intelligence and neuroscience, vol. 2021, pp. 1-14, 2021.

[39] J. Luo et al., "Container-based fog computing architecture and energy-balancing scheduling algorithm for energy IoT," Future Generation Computer Systems, vol. 97, pp. 50-60, 2019.

[40] I. Attiya, L. Abualigah, D. Elsadek, S. A. Chelloug, and M. Abd Elaziz, "An intelligent chimp optimizer for scheduling of IoT application tasks in fog computing," Mathematics, vol. 10, no. 7, p. 1100, 2022.

[41] W. Liu, C. Li, A. Zheng, Z. Zheng, Z. Zhang, and Y. Xiao, "Fog computing resource-scheduling strategy in IoT based on artificial bee colony algorithm," Electronics, vol. 12, no. 7, p. 1511, 2023.

[42] M. Kaur, R. Sandhu, and R. Mohana, "A framework for scheduling IoT application jobs on fog computing infrastructure based on QoS parameters," International Journal of Pervasive Computing and Communications, vol. 19, no. 3, pp. 364-385, 2023.

[43] S. Mousavi, S. E. Mood, A. Souri, and M. M. Javidi, "Directed search: a new operator in NSGA-II for task scheduling in IoT based on cloud-fog computing," IEEE Transactions on Cloud Computing, 2022.

[44] I. Attiya, M. Abd Elaziz, and I. Issawi, "An improved hunger game search optimizer based IoT task scheduling in cloud-fog computing," Internet of Things, p. 101196, 2024.

[45] J. Z. Ahmadabadi, S. E. Mood, and A. Souri, "Star-quake: A new operator in multi-objective gravitational search algorithm for task scheduling in IoT based cloud-fog computing system," IEEE Transactions on Consumer Electronics, 2023.

[46] N. A. Alsamarai, O. N. Uçan, and O. F. Khalaf, "Bandwidth-deadline IoT task scheduling in fog–cloud computing environment based on the task bandwidth," Wireless Personal Communications, pp. 1-17, 2023.

[47] A. Azadi and M. Momayez, "Review on Constitutive Model for Simulation of Weak Rock Mass," Geotechnics, vol. 4, no. 3, pp. 872-892, 2024, doi: https://doi.org/10.3390/geotechnics4030045.

[48] M. D. Tezerjani, M. Khoshnazar, M. Tangestanizadeh, and Q. Yang, "A Survey on Reinforcement Learning Applications in SLAM," arXiv preprint arXiv:2408.14518, 2024, doi: https://doi.org/10.48550/arXiv.2408.14518.

[49] M. Bhatia, S. K. Sood, and S. Kaur, "Quantumized approach of load scheduling in fog computing environment for IoT applications," Computing, vol. 102, no. 5, pp. 1097-1115, 2020.

[50] H. S. Ali, R. R. Rout, P. Parimi, and S. K. Das, "Real-time task scheduling in fog-cloud computing framework for iot applications: A fuzzy logic based approach," in 2021 International Conference on COMmunication Systems & NETworkS (COMSNETS), 2021: IEEE, pp. 556-564.

[51] J. Lim, "Latency-aware task scheduling for IoT applications based on artificial intelligence with partitioning in small-scale fog computing environments," Sensors, vol. 22, no. 19, p. 7326, 2022.

[52] R. O. Aburukba, T. Landolsi, and D. Omer, "A heuristic scheduling approach for fog-cloud computing environment with stationary IoT devices," Journal of Network and Computer Applications, vol. 180, p. 102994, 2021.

[53] 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.

[54] 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, 2023.

[55] M. Goudarzi, M. A. Rodriguez, M. Sarvi, and R. Buyya, "$\mu $-DDRL: A QoS-Aware Distributed Deep Reinforcement Learning Technique for Service Offloading in Fog computing Environments," IEEE Transactions on Services Computing, 2023.

[56] J. Lu et al., "A2C-DRL: Dynamic Scheduling for Stochastic Edge-Cloud Environments Using A2C and Deep Reinforcement Learning," IEEE Internet of Things Journal, 2024.

[57] Z. Wang, M. Goudarzi, M. Gong, and R. Buyya, "Deep Reinforcement Learning-based scheduling for optimizing system load and response time in edge and fog computing environments," Future Generation Computer Systems, vol. 152, pp. 55-69, 2024.