SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Critical Evaluation Of Scheduling And Ranking Algorithms In Fog-Cloud Computing

Mohammed Mohiuddin Qadri Syed[1] , Pachipala Yellamma[2]

[1]Research Scholar Department Of Computer Science and Engineering Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India. smmqadriin@gmail.com
[2]Department Of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India. pachipala.yamuna@gmail.com

Abstract
The rapid growth of fog and cloud computing has created a need for the development and implementation of intelligent scheduling and ranking algorithms to improve the allocation and execution efficiency. In the study, the efficiency and evaluation of different heuristic, metaheuristic, and artificial intelligence-based scheduling and ranking algorithms, such as First Come First Serve (FCFS), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Deep Reinforcement Learning (DRL), in the context of fog and cloud computing environments, have been explored. The experimental study reveals that the proposed DRL-based approach has better efficiency and the minimum makespan, i.e., 61.3 seconds, for the execution of 500 tasks, compared to the traditional FCFS approach, which has a makespan of 84.5 seconds. In terms of energy consumption, the proposed approach has the minimum energy consumption, i.e., 454.6 joules, compared to the traditional FCFS and GA approaches, which have energy consumption of 679.7 joules and 567.7 joules, respectively. The proposed approach has the highest CPU utilization, i.e., 78.6%, and the least idle time, i.e., 12.3%, for the execution of tasks, and the minimum average response time, i.e., 119.5 milliseconds. These results demonstrate the limitations of traditional approaches in dynamic fog-cloud environments and verify the potential benefits of adaptive AI-driven models in enhancing efficiency in scheduling, reducing latency in scheduling, and saving power in scheduling. These results provide a foundation for the design of sophisticated scheduling strategies to meet the needs of real-time distributed computing environments. These results demonstrate the limitations of traditional approaches in the dynamic fog-cloud environment and verify the potential benefits of the proposed adaptive AI-driven models to efficiently tackle the problem of scheduling efficiency, latency reduction, and power saving. These results provide a foundation for the design of sophisticated scheduling strategies to meet the needs of real-time distributed computing environments.

Keywords: Fog Computing, Scheduling Algorithms, Heuristic Optimization Techniques, Reinforcement Learning, Resource Management, Cloud-Fog Environment.

1. Introduction

The exponential growth of Internet of Things (IoT) technologies with respect to time has led to a significant rise in the amount of real-time data being processed by smart devices. These devices require fast and efficient data processing capabilities to enable applications such as healthcare monitoring and smart traffic control. Although cloud computing provides devices with computing and storage capabilities, there is a physical distance between devices, which causes latency, network congestion, and a high energy overhead, making it less suitable for real-time applications [1]. To address these challenges, fog computing has been proposed as an extension of the traditional cloud computing paradigm, which provides decentralized infrastructure with computing, storage, and networking services closer to the source of data. Fog nodes, such as edge routers, gateways, and microservers, help to process data closer to the source, which improves the latency, throughput, and real-time response of data. Fog computing with the support of cloud infrastructure provides a hierarchical architecture for task execution, which needs to be dynamically managed between the fog and cloud layers [1, 2].

The hybrid fog-cloud infrastructure is also associated with challenges related to task scheduling and resource ranking. Scheduling is critical to ensure timely execution, load balancing, energy efficiency, and Quality of Service (QoS) constraints such as makespan, cost, latency, and resource utilization [1]. The inherent heterogeneous and dynamic nature of the fog-cloud architecture requires the development of adaptive scheduling techniques to function efficiently within different network conditions and constraints. A number of scheduling techniques have been proposed in the literature over the past few years. The use of heuristic scheduling techniques, such as First Come First Serve (FCFS) and Shortest Job First (SJF), is easy to implement, but they may not function efficiently within dynamic environments [2]. Similarly, the use of metaheuristic techniques, such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO), has been reported to function more efficiently for task scheduling in the fog environment by exploring the search space more extensively [1]. Recently, the use of AI-based scheduling techniques, such as Deep Reinforcement Learning (DRL), has been reported to function more efficiently for task scheduling in the fog environment by dynamically adapting to the changing environment [3, 4, 5]. The proposed model has the potential to function more efficiently by dynamically adapting to the changing environment [6].

In an extensive study on task scheduling techniques in fog computing, Alizadeh et al. [2] highlighted the need for more intelligent and context-aware task scheduling techniques. Rahimikhanghah et al. [3] also identified different task scheduling models and proposed an adaptive framework to resolve delay and inefficiency problems. Bansal et al. [1] presented an extensive classification of task scheduling techniques into four categories: static, dynamic, heuristic, and hybrid, depending on the performance objectives and application domains. Wang et al. [6] demonstrated the efficiency of DRL-based task scheduling in minimizing the load and response time of tasks in fog-edge networks compared to traditional task scheduling models. Subramoney and Nyirenda [7] also conducted a comparative evaluation of population-based optimization techniques and concluded that metaheuristic algorithms perform better than heuristic algorithms but lag behind DRL-based task scheduling techniques in terms of adaptability. Anka et al. [8] proposed a bio-inspired task scheduling model for fog-cloud environments with the objective of optimizing energy and delay constraints. Ijaz et al, [9] proposed an improved version of the multi-objective differential evolution algorithm for task scheduling in fog environments.

Although there is a vast amount of literature on the subject, there is a lack of empirical comparison of the different methods using a unified framework. The objective of the current study is to bridge the gap by critically evaluating and comparing the different heuristic, metaheuristic, and AI-based methods in task scheduling using a simulated environment. The performance metrics used in evaluating the different methods are makespan, energy, resources, and response time. The findings of the research will provide valuable insights into the different task scheduling frameworks. By identifying the performance bottlenecks and the better behaviors of the different approaches, the research aims to develop the foundation for the development of the next generation of task scheduling approaches that meet the needs of the modern fog-cloud environment

2. Related Works

In the last ten years or so, different researchers have worked to address the intricacy of scheduling in fog-cloud ecosystems, introducing different models to ensure the optimization of Quality of Service (QoS) as well as efficiency and response times. Some of the contributions to the field of fog-cloud ecosystems scheduling include the introduction of a semi-greedy algorithm to ensure the optimization of deadline-constrained as well as energy-constrained IoT tasks in fog-cloud systems by Azizi et al. [10]. Another contribution to the field was the introduction of a cost-effective model to ensure the optimization of critical heartbeat tasks in medical IoT systems through fog-cloud ecosystems by Mastoi et al. [11]. Talaat et al., on the other hand, introduced a hybrid model to ensure the optimization of critical tasks through the use of a combination of Convolutional Neural Networks (CNN) as well as Modified Particle Swarm Optimization (MPSO) to ensure the optimization of delay sensitivity as well as load balancing through fog-cloud ecosystems [12]. In the context of AI-based scheduling, Javanmardi et al. introduced the FPFTS framework, which combines fuzzy logic and mobility-aware PSO. This approach has the advantage of adapting to dynamic topological changes in the network and minimizing the overall application loop delay and fog network congestion. In a similar vein, Lin et al. introduced a distributed fog computing model for vehicular networks, leveraging the advantages of Software-Defined Networking (SDN) and the efficacy of combined delay optimization and topology-aware scheduling. The authors demonstrated the advantages of the proposed model by leveraging the combined advantages of delay optimization and topology-aware scheduling [14]. Najafizadeh et al. introduced a goal programming-based multi-objective scheduling model, where the trade-offs between latency, energy, and SLA violations were addressed. The experimental framework for the proposed model demonstrated its efficacy in minimizing deadline violations and energy consumption [15]. Another notable contribution by Madhura et al. was the improved list-based scheduling technique, where the proposed model demonstrated its efficacy by categorizing and prioritizing tasks based on dynamic queue strategies, thus improving the overall makespan reduction and cost efficiency over traditional models like HEFT and PEET [16]. On the metaheuristic side, Potu et al. implemented the extended Particle Swarm Optimization model to tune the inertia weights as well as the cognitive coefficients with real-time parameters of the tasks. The results indicated the efficiency of the proposed model with reduced overall execution times as well as efficiency in the mapping of tasks to the nodes compared to the traditional approaches [17]. El-Nattat et al. proposed a deadline-constrained scheduling algorithm with enhanced efficiency compared to traditional heuristics in delay- as well as cost-sensitive scenarios of fog-cloud architectures, such as smart city monitoring systems [18]. These research contributions indicate the evolutionary trend from traditional heuristics to intelligent algorithms with adaptability to the fog-cloud architecture. However, the research community is yet to fill the gap with the design of efficient algorithms with the ability to balance the response times with the efficiency of the heterogeneous tasks as well as the resource availability. This research contribution aims to present a comparative analysis of the efficiency of the traditional heuristic, metaheuristic, as well as the AI-based DRL algorithms with the help of a common simulation platform to present quantification of the real-time efficiency of the algorithms with the fog-cloud architecture.

Another direction that has been explored in recent literature is the use of workflow-based and application-specific scheduling frameworks. For example, Dubey et al. proposed a QoS-aware workflow scheduling framework for fog environments in the context of scientific data, which included the handling of deadlines and node availability in real time to maximize the overall throughput and minimize SLA violations. The study demonstrated the efficiency of dependency-aware scheduling for improving the efficiency of applications like weather simulation and video analytics [19]. In the case of delay-sensitive scheduling, for instance, Xu et al. presented a cooperative scheduling strategy for task offloading in vehicular fog networks. The proposed strategy dynamically schedules the fog nodes according to the density of vehicles and the deadline window of task execution. This strategy increases the responsiveness of the overall system in a congested environment [20]. Another novel strategy is the hybrid approach that combines Software Defined Networking (SDN) and Machine Learning (ML). In this regard, Ahmad et al. presented a novel strategy for scheduling in the fog-cloud environment in the field of smart agriculture. The proposed model integrated a predictive load balancing method using a supervised learning method. This increased the overall adaptability of the network as well as reduced the overall delay due to the queueing phenomenon [21].

Although many algorithms have been proposed in the field of fog networks, there is a lack of proper testing using standardized simulators. This reduces the efficiency of the proposed algorithms. In the proposed paper, the authors have utilized the iFogSim simulator and the CloudSim simulator to provide a standardized environment for testing the proposed heuristic, metaheuristic, and AI-based algorithms. Many researchers have utilized the iFogSim simulator; however, some have utilized a custom-built simulator. Moreover, most of the existing approaches have focused mainly on one performance metric, i.e., either energy consumption or latency, without considering both factors. Hence, there is a pressing need for developing intelligent schedulers that can optimize more than one QoS parameter without compromising scalability.

Table 1 presents a consolidated comparison of some of the most popularly employed scheduling algorithms, which have found wide acceptance in the context of fog-cloud computing, categorized into three classes, namely, heuristic, metaheuristic, and AI-based scheduling algorithms. Each scheduling algorithm is assessed from the perspectives of their primary optimization goals, quality of service parameters, simulation tools employed, and their performance advantages. As shown in Table 1, heuristic algorithms such as FCFS and EAM are simple algorithms with low complexity, which is appropriate for environments with low variation. However, their poor adaptability results in poor performance when the task is high or the network is changing. For example, the poor resource utilization of the FCFS algorithm results in poor delay response, while the EAM algorithm performs better than the FCFS algorithm with average makespan consideration.

Table 1: Comparative Evaluation of Fog-Cloud Scheduling Algorithms

Algorithm	Category	Optimization Focus	QoS Parameters Considered	Simulation Tool	Performance Benefit
FCFS	Heuristic	Task Order	Makespan, Latency	iFogSim	Easy to implement but inefficient under load

Algorithm	Category	Optimization Focus	QoS Parameters Considered	Simulation Tool	Performance Benefit
GA	Metaheuristic	Population Search	Makespan, Cost, Latency	CloudSim/iFogSim	Balances load better, medium convergence speed
PSO	Metaheuristic	Swarm Intelligence	Makespan, Energy, Throughput	CloudSim	Lower energy usage, better resource allocation
DRL	AI-Based	Policy Learning	Latency, Energy, Response Time, Utilization	iFogSim	Highest adaptability and overall performance
EAM	Heuristic	Deadline-Aware Avg Makespan	Average Makespan, Throughput, Response Time	iFogSim	Outperforms RR for real-time systems
EPSO	Metaheuristic	Enhanced Particle Swarm	Makespan, Load Balancing	iFogSim	Superior convergence, better node distribution
CNN+MPSO	Hybrid AI-Opt	Load Prediction + Task Mapping	Load Balancing, Delay, Energy	CloudSim/iFogSim	Best balance between prediction and scheduling
FPFTS	Fuzzy-AI Hybrid	Delay-Aware + Node Mobility	Application Loop Delay, Resource Availability	iFogSim	Effective in mobile fog scenarios

Meta-heuristics, as represented by GA and PSO, provide more flexible optimization. The former uses evolutionary intelligence, while the latter uses swarm intelligence for task allocation. Both algorithms have shown improvements over the use of heuristics, as represented by their energy efficiency and task allocation characteristics. However, they require parameter tuning, and their convergence rates depend on the complexity of the workload. The improved versions, as represented by EPSO, have refined this approach by adding stability to the convergence characteristics and more efficient task allocation. The use of DRL, as well as other hybrid approaches like CNN+MPSO or FPFTS, is clearly superior to other approaches, as represented by their flexibility and efficiency. The former, for instance, can learn the optimal policies for task scheduling through continuous performance feedback. In the case of CNN+MPSO, the model uses predictive analytics as well as optimization to ensure accurate predictions of the loads to be handled, followed by intelligent task mapping. FPFTS is also able to effectively manage delays occurring to the application in the mobile fog environment through the incorporation of fuzzy decision-making.

All the above-mentioned simulation tools, such as iFogSim and CloudSim, play a significant role in validating the above models under standardized conditions. Overall, the above comparative table demonstrates the limitations of static scheduling approaches as well as the requirement for intelligent algorithms to manage the complexities of the fog-cloud environment effectively. The trend is moving away from rule-based scheduling to complex learning algorithms to meet the requirements of the fog-cloud environment effectively. This study is a part of the trend as the authors have implemented the algorithms to evaluate their real-world application scenarios effectively.

3. Methodology

This research paper follows a structured and multi-phase research methodology to analyze, simulate, and evaluate different algorithms for scheduling and ranking in the context of fog-cloud computing. The research methodology includes the analytical classification of algorithms, mathematical modeling of the performance objectives, experimental evaluation through the simulation-based data set, comparative performance evaluation, and the limitations of the research methodology.

Analytical Classification of Algorithms: The algorithms used in this research paper have been classified into three main categories according to the logic and optimization strategy used by the algorithms. The first set of algorithms includes heuristic algorithms like First Come First Serve (FCFS), Shortest Job First (SJF), and

Min-Min algorithms, which work according to a set of predefined rules using simple decision-making criteria. The second set includes metaheuristic algorithms like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO), which work on the principle of population-based optimization and avoid local optima. The third set comprises AI-based algorithms such as Deep Reinforcement Learning (DRL), Q-Learning, and Fuzzy Logic, which operate based on adaptive policies.

These algorithms were chosen to represent a wide range of scheduling behaviors. They were also tested based on their scalability, energy efficiency, fairness in task allocation, and their adaptability to environmental changes in a fog/cloud system.

Mathematical Modeling and Performance Metrics: The problem of scheduling can be mathematically represented as a multi-objective optimization problem. The objectives are to minimize makespan (Makespan refers to the total time required to complete all the scheduled tasks in a system), minimize energy consumption, and maximize resource utilization. Makespan can be represented as:

$$\min\left(\max_{i\in\{1,2\ldots n\}} C_i\right)$$

where C_i denotes the completion time of the i^{th} task. The energy consumption of computing nodes is modeled as:

$$E = \sum_{j=1}^{m}\left(P_j^{idle}.T_j^{idle} + P_j^{active}.T_j^{active}\right)$$

where P_j is the power usage and T_j is the time duration in idle or active mode for node j. Resource utilization is calculated by:

$$U = \frac{\sum_{j=1}^{m} BusyTime_j}{\sum_{j=1}^{m} TotalTime_j}$$

This helps to quantitatively evaluate each of the algorithms under a similar performance evaluation criterion. The experimental data used for this study is simulated using a simulation-based modeling technique provided by iFogSim and CloudSim tools. These are highly popular tools for simulating fog/cloud infrastructures, which are capable of workload behavior emulation, as depicted by various studies such as Alizadeh et al. (2020), Bansal et al. (2022), Rahimikhanghah et al. (2021) [1-3].

The simulated dataset comprises a variable number of tasks ranging from 100 to 500 each with randomly assigned computational loads between 100 and 10,000 million instructions (MI). Fog nodes, configured between 10 and 30 units, represent resource-constrained edge devices with limited CPU, memory, and energy capacity. The cloud layer contains high-performance virtual machines capable of handling overflow workloads. Task arrivals follow a Poisson distribution to mimic dynamic IoT behavior. Network links between fog and cloud layers include latency settings of 1–10 ms and 50–150 ms respectively, while bandwidth ranges from 100 Mbps to 1 Gbps. The architecture of the three-tier system used in the simulations is depicted in Figure 1. In the architecture, IoT devices, fog nodes, and cloud servers have been used.

Figure 1: Fog-Cloud Simulation Architecture with Task Flow Distribution

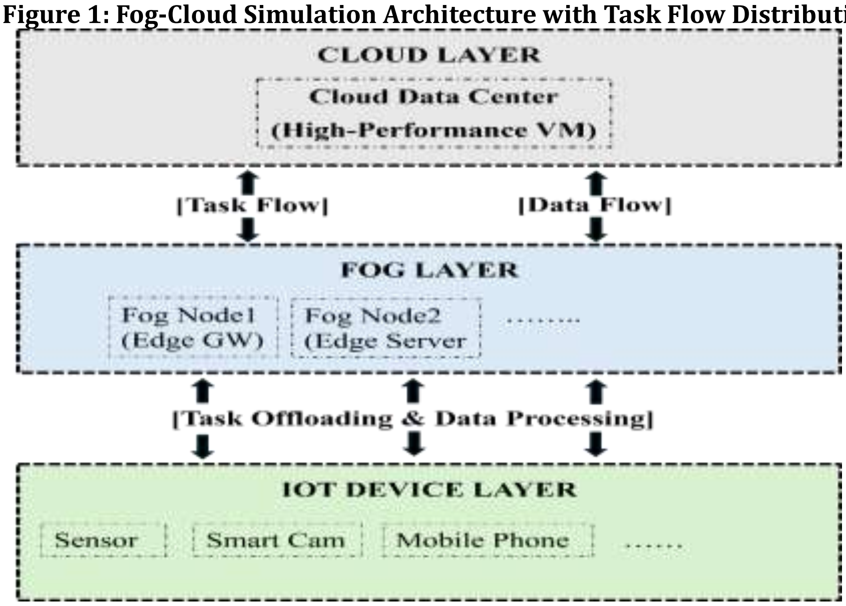

Figure 1, Fog Cloud Simulation Architecture shows a layered representation of the operational workflow in a fog-cloud computing environment, which is specifically designed to illustrate the data flow and task scheduling principles used in this research. At the lowest layer of the architecture is the IoT Device Layer, which consists of end-user devices such as environmental sensors, surveillance cameras, and smart mobile devices. These devices produce a constant flow of data and tasks that need to be processed. However, the devices' limited processing capabilities mean that tasks need to be offloaded to other layers for execution. Moving up the layers, the Fog Layer is responsible for the initial processing of tasks and decision-making close to the data source or device. This layer consists of intermediate nodes such as edge gateways, local servers, routers, and micro data centers. This layer is considered to have moderate processing and storage capabilities, making it efficient in processing tasks with low latency.

The Cloud Layer is at the highest layer of this architecture. This layer is comprised of centralized data centers and virtual machines that have high computational power. This layer is responsible for processing complex computational tasks, analysis of historical data, and storage-intensive tasks that cannot be processed at the fog nodes. Any task that is considered too large or not critical in terms of execution time is offloaded from the fog layer to the cloud layer for execution. The flow of tasks in this architecture starts from the IoT layer and proceeds vertically upwards to the fog nodes for execution. In case the task cannot be executed at the fog nodes or the nodes do not have the capability to execute the task, it is escalated to the cloud layer. Conversely, the flow of data, which is the result or response to a task, flows downwards from the initiating IoT device. This architectural model is useful for achieving the objectives of the research work as it provides an environment for simulating and evaluating scheduling algorithms on a multi-tiered architecture.

Evaluation Tools and Experimental Setup: All simulations are conducted using iFogSim for fog-dominant scenarios and CloudSim for centralized clouds. For each test scenario, 5 to 10 fog clusters with one or more cloud data centers are used. Fog devices are set with initial values for their parameters, such as CPU frequency (0.5 to 1.5 GHz), RAM size (256 to 1024 MB), and energy budget limit. The simulated environment is set with constraints such as thermal constraints, power constraints, and task generation rates. All algorithms are simulated with similar settings to ensure fairness. Additionally, the simulations are run 30 times to reduce random effects.

Comparative Analysis Approach: A normalized scoring method is used to ensure fairness and consistency in comparing the results of the algorithms. For a set of result metrics X, the normalized score is calculated using:

$$\text{Normalized Score}_{\mathrm{i}} = \frac{\mathrm{X_i} - \mathrm{X_{min}}}{\mathrm{X_{max}} - \mathrm{X_{min}}}$$

Weighted scores are then calculated using significance factors for each metric, with more importance being assigned to latency and energy metrics, as these are more relevant in edge computing scenarios. The overall ranking of each algorithm is calculated using the weighted average of scores across all metrics. This is a comprehensive approach to comparing algorithms using multiple metrics, which also helps in finding the best balanced scheduling models for fog-cloud scenarios.

Limitations and Assumptions: Though the experimental setup is comprehensive, there are a few assumptions and limitations to the study. For instance, though the workload is generated using a synthetic model, it is assumed that it accurately represents the workload generated by IoT devices. Additionally, there is no consideration of task interdependencies in the form of a directed acyclic graph (DAG) in the simulations. Thirdly, the energy consumption model is assumed to be constant, whereas in practice, it varies with temperature and voltage scaling. Fourthly, though the use of AI-based schedulers is explored, it is assumed that the convergence of the algorithms is not affected in an unpredictable fashion.

Future Work: All the assumptions and limitations will be considered while further enhancing the proposed algorithmic evaluation approach.

4. Results and Discussion

In order to evaluate the efficiency of various scheduling and ranking approaches in the context of fog-cloud computing environments, various experiments are performed based on the results obtained from the simulation-based data sets generated by iFogSim and CloudSim tools. The results obtained from the experiments are presented in the following figure, showing the makespan values obtained by implementing various scheduling approaches, including FCFS, GA, PSO, and DRL, while varying the task load from 100 to 500.

Figure 2: Makespan values obtained by implementing various scheduling approaches

As presented in the figure above, the makespan values obtained by implementing the FCFS approach are found to be the highest among all the scheduling approaches, while the makespan values obtained by implementing the DRL approach are found to be the lowest among all the scheduling approaches. For instance, while implementing the FCFS approach, the makespan value obtained by scheduling 500 tasks is found to be 84.5 seconds, while the makespan value obtained by implementing the DRL approach is found to be 61.3 seconds. The significant reduction in the time taken for execution is a result of the ability of DRL to allocate tasks adaptively according to the availability of resources. The performance of metaheuristic algorithms, i.e., GA and PSO, is moderate, as they maintain a lower value of makespan, though they lag behind DRL.

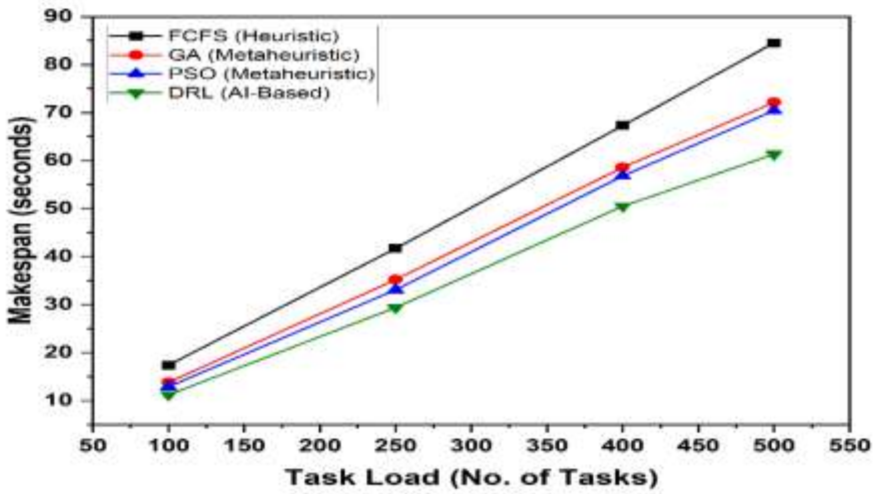

Figure 2: Makespan Comparison Chart for Various Scheduling Algorithms

PSO is slightly better than GA across all the task loads, especially at 250 and 400 tasks, where the difference is greater. This shows the better exploration capacity of the search space with the PSO algorithm compared to the GA algorithm. However, the results confirm the better scalability and efficiency of the DRL-based scheduling algorithm compared to the others. These results are quite significant in the context of the fog-cloud system, which is required to handle high-frequency data streams.
Figure 3 shows the comparison of the energy consumption of the fog-cloud system with different scheduling algorithms like FCFS, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Deep Reinforcement Learning (DRL). The results are shown with the unit of energy consumption as joules, which includes the energy consumption of the fog node, the cloud server, and the entire system. The analysis shows the highest energy consumption with the FCFS algorithm at 679.7 joules with the highest consumption at the cloud server at 496.1 joules due to inefficient offloading of tasks to the cloud server because of poor handling at the fog layer.

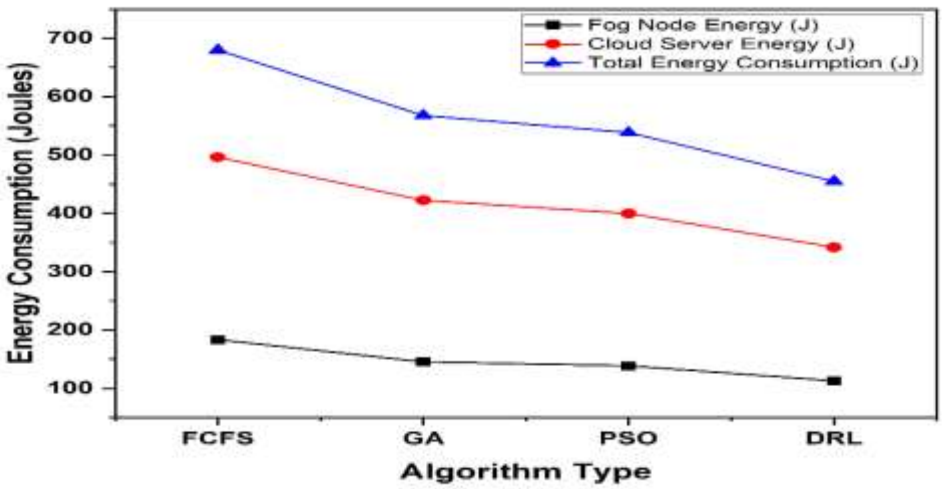

Figure 3: Energy Consumption Distribution across Scheduling Algorithms

On the other hand, the proposed DRL-based scheduler presents the most energy-efficient characteristics, consuming the least total energy, i.e., 454.6 joules. It is noteworthy that the proposed algorithm achieves the least energy consumption for the fog nodes, i.e., 112.9 joules, and the cloud server, i.e., 341.7 joules. This implies that the proposed algorithm maintains a higher percentage of tasks in the fog nodes. In contrast, the GA and PSO algorithms exhibit moderate energy efficiency. Among the two algorithms, the PSO algorithm presents slightly better energy efficiency characteristics. The PSO algorithm achieves the least total energy consumption, i.e., 538.4 joules, compared to the GA algorithm, i.e., 567.7 joules. In addition, the proposed PSO algorithm achieves the least energy consumption for the fog nodes and the cloud server. This implies that the PSO algorithm presents better task allocation characteristics. In conclusion, the figure demonstrates the superiority of the DRL model, which is a significant factor in sustaining long-term operation in fog-cloud computing, particularly considering the power constraints associated with edge devices. The results affirm that not only do intelligent schedulers improve performance, but they are also significant in reducing energy consumption in distributed computing layers.

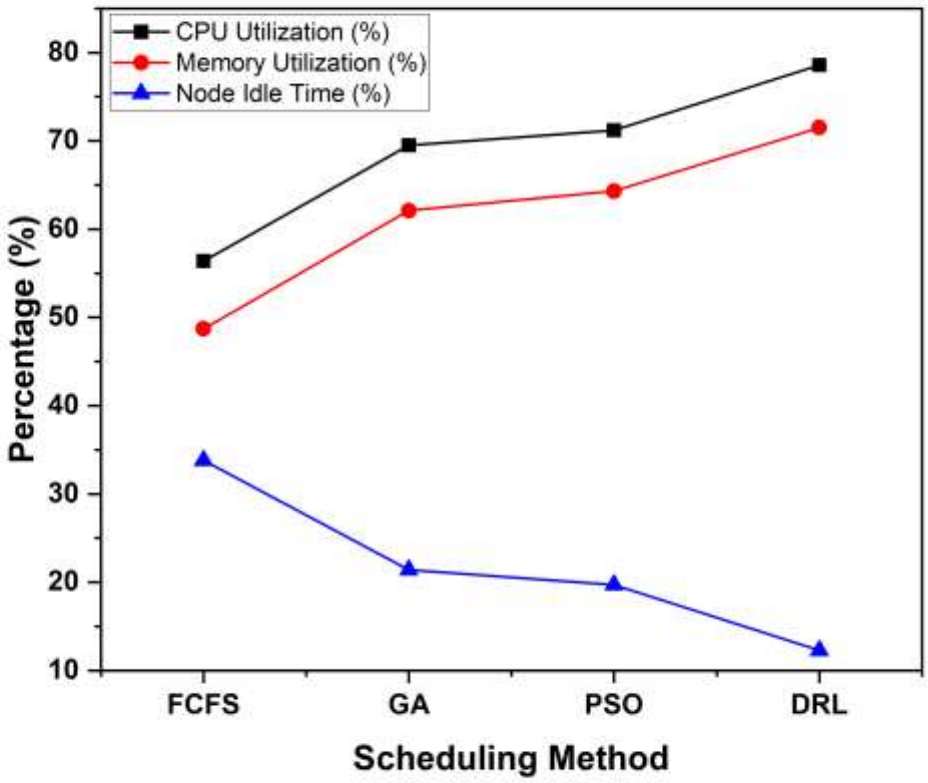

Figure 4: Fog Node Resource Utilization by Different Scheduling Algorithms

Figure 4 depicts the average level of resource utilization by four different scheduling strategies, namely FCFS, GA, PSO, and DRL, on the fog node, for three different parameters, namely CPU, memory, and idle time. It is evident from the graph that the proposed DRL-based scheduling approach has the highest level of CPU and memory utilization, i.e., 78.6% and 71.5%, respectively, and the least idle time, i.e., 12.3%. This indicates the efficiency and efficacy of the proposed approach in involving the fog node in the task allocation process. The intelligent decision-making capability of the proposed approach based on the DRL technique allows for better matching of the task requirements and the node capabilities. Among the two metaheuristic-based approaches, the proposed PSO technique has a higher level of CPU and memory utilization, i.e., 71.2% and 64.3%, respectively, and the least idle time, i.e., 19.7%, compared to the GA technique, which has 69.5% CPU, 62.1% memory, and 21.4% idle time. These results indicate that PSO can effectively distribute the tasks with greater uniformity across the fog layer infrastructure to minimize unproductive times as well as balance the load efficiently. On the other hand, the results of the FCFS algorithm indicate poor resource engagement with 56.4% CPU utilization and 48.7% memory utilization. Moreover, the 33.8% idle times of the FCFS algorithm are the highest compared to the other approaches. These results indicate poor scheduling decisions made by the FCFS algorithm, which leads to poor utilization of the CPU capacity of the fog layer infrastructure. These results clearly indicate the efficiency of the AI-based scheduling approach, i.e., DRL, to engage the fog layer resources efficiently.

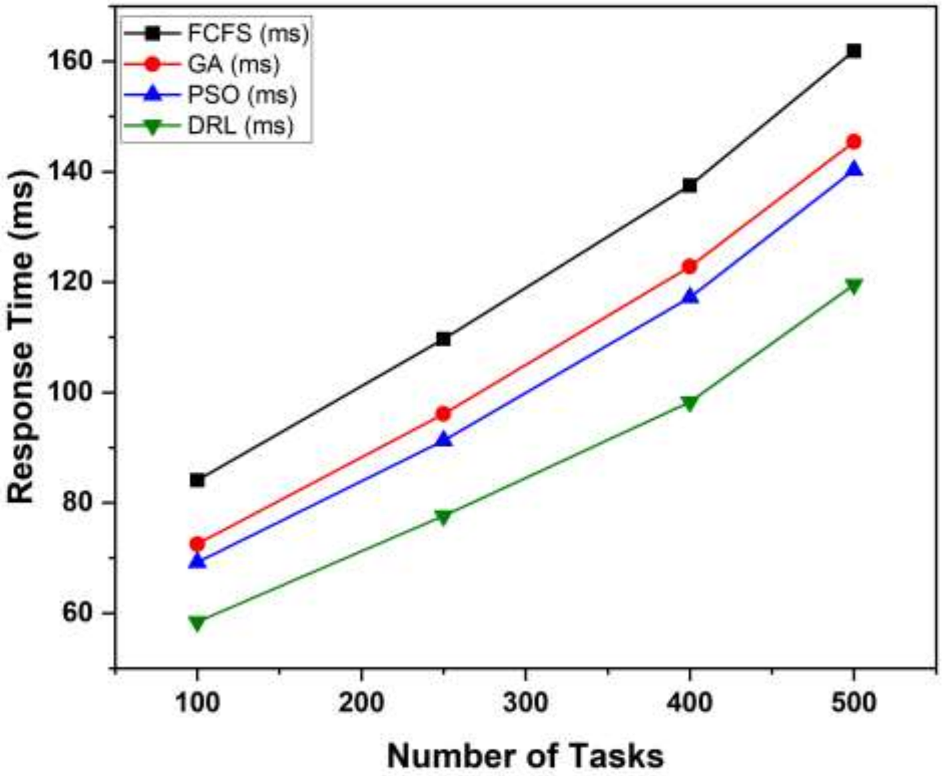

Figure 5: Response Time Analysis for IoT Task Completion

Figure 5 illustrates a comparative study on the average response time measured in milliseconds for four different algorithms: FCFS, GA, PSO, and DRL. In this context, the response time is defined as the time interval between the arrival of the task at the fog-cloud platform and the initiation of the task's execution. This parameter is considered more important in IoT-centric applications, as the response time affects the quality of service. In all cases, the proposed DRL-based task scheduling model achieves the lowest response time, which indicates the efficient handling of arriving tasks. For instance, at a task load of 500, DRL results in a response time of 119.5 ms, making it better than FCFS by more than 40 milliseconds. This is a significant improvement, especially in latency-critical applications. The PSO algorithm comes second, as it maintains response times lower than those of GA and FCFS at all times. Under the maximum task load of 500, PSO results in a response time of 140.3 ms, while GA takes 145.4 ms and FCFS takes 161.9 ms. This shows the superiority of PSO in clearing queue backlogs. GA has a moderate performance, as it offers improvements over the baseline heuristic but lacks the responsiveness of PSO and DRL. FCFS, however, has the highest response times at all task load levels, including the lowest of 100, at which it takes 84.1 ms. Its non-adaptive nature in linear scheduling logic makes it difficult in meeting the concurrent demands of the IoT environment. The data collected in this section further reiterates the efficiency of DRL in reducing latency in the system. This way, it becomes efficient in time-bound applications such as smart surveillance, industrial control, and healthcare in a fog-cloud computing system.

Discussion: The results obtained in the experiments of this study clearly reflect the significance of the performance differences between various scheduling algorithms in a fog-cloud computing system. The performance evaluations of the scheduling algorithms are based on four significant parameters: makespan, energy consumption, resource utilization, and response time. All these parameters are significant in ensuring quality of service in distributed computing environments supporting IoT applications. One of the most important observations is the makespan reduction capability of the Deep Reinforcement Learning (DRL) approach. Thus, as the volume of tasks was increasing, DRL was consistently showing superior task completion efficiency. The adaptive scheduling feature helped the system to map tasks to appropriate nodes, thus minimizing task execution delays. When compared with the conventional approach, i.e., FCFS, which was not efficient in handling a large volume of tasks, the performance of DRL was found to improve proportionally with the volume of tasks, thus establishing the robustness and flexibility of DRL.

In terms of energy efficiency, DRL was found to be the most efficient. By reducing unnecessary task offloading to distant cloud servers and keeping tasks at the fog layer whenever possible, DRL was found to minimize both node-level and system-level energy consumption. The other algorithms, particularly FCFS, were found to consume more energy due to inefficient resource allocation. The metaheuristic algorithms, i.e., GA and PSO, showed a reasonable performance but lagged behind DRL in terms of energy consumption. It sustained the maximum CPU and memory engagement. Also, the node idle time remained low. This efficient usage of resources does not only help in the faster processing of tasks but also reduces the chance of resource bottlenecks. The static behavior of the FCFS algorithm resulted in high idle time for the fog nodes. This implies low load balancing and task allocation efficiency. GA and PSO showed similar efficiency in resource usage but were unable to show the dynamic behavior of the DRL algorithm in different network conditions. Among all the response time parameters, the significance of the DRL algorithm was highlighted. The reduced response time from the arrival of the task to the start of the task execution was quite high compared to the other algorithms. This is a vital feature, especially in real-time applications. This feature is more critical in IoT-based applications. Based on the above discussion, it is quite clear that the AI-based algorithm, especially the DRL algorithm, is the most efficient algorithm in optimizing the performance of the fog-cloud infrastructure. The smart learning capabilities, smart tuning, and flexibility in adapting to changing resources of the DRL algorithm make it more efficient compared to the other algorithms. Although the performance of the PSO and GA algorithms is more efficient compared to the traditional algorithms, it is handicapped in adapting the control parameters and the rate of convergence. The DRL algorithm, being a dynamic algorithm, can guarantee the optimal performance even in a changing environment.

The results achieved in this paper are a clear proof of the necessity to incorporate learning-based models in the development of the next generation of fog/cloud computing systems. The enhancements achieved in makespan, energy consumption, resource allocation, and responsiveness are clear evidence of the necessity to incorporate learning-based models in the development of efficient and sustainable distributed computing systems. The future scope of this work can be directed to incorporating real-world data streams and task dependencies along with hybrid AI to further enhance the efficiency of the proposed approach.

5. Conclusion

In this research, the performance of various scheduling and ranking algorithms was evaluated with regards to their efficiency in managing the dynamic workload of IoT application systems in fog-cloud environments through the use of simulations. The algorithms considered included heuristic (FCFS), metaheuristic (GA & PSO), and AI-based algorithms (Deep Reinforcement Learning). The algorithms were evaluated with regards to four primary metrics or criteria: makespan, energy consumption, resource utilization, and response times. The results of the research show conclusively that the DRL algorithm was the best compared to the other algorithms with regards to all the metrics considered. For instance, the DRL algorithm was the best with regards to the makespan metric, having the least value of 61.3 seconds compared to the 84.5 seconds of the FCFS algorithm and 72.1 seconds of the GA algorithm when the system was subjected to a 500-task load. With regards to the energy consumption metric, the DRL algorithm was the best with a value of 454.6 joules compared to 679.7 J of the FCFS algorithm and 567.7 J of the GA algorithm. Furthermore, the DRL ensured the optimal efficiency of resources, with CPU utilization at 78.6% and memory utilization at 71.5%, whereas the idle time was minimized to only 12.3%. Also, the proposed model ensured the fastest task responsiveness, with the average response time being only 119.5 ms when the tasks were at their maximum load, compared to 161.9 ms when the FCFS algorithm was used. From the above observations, it is evident that the proposed DRL-based scheduling technique is extremely efficient in terms of optimal allocation of resources, minimal delays in task execution, and lower energy requirements in the proposed HFGC architecture. Even though the GA and PSO algorithms were satisfactory in terms of efficiency, they were still lagging when compared to the proposed technique. In conclusion, the proposed technique of incorporating intelligent scheduling mechanisms like DRL is of utmost importance in the efficient functioning of the proposed HFGC architecture. By greatly improving system responsiveness and sustainability, these models have great potential for being used in real-time, resource-constrained IoT systems. Future work will be focused on the integration of dependency-aware task models, mobility scenarios, as well as real-world deployment conditions.

References:

1. Bansal, S., Aggarwal, H. and Aggarwal, M., 2022. A systematic review of task scheduling approaches in fog computing. Transactions on Emerging Telecommunications Technologies, 33(9), p.e4523.

2. Alizadeh, M.R., Khajehvand, V., Rahmani, A.M. and Akbari, E., 2020. Task scheduling approaches in fog computing: A systematic review. International Journal of Communication Systems, 33(16), p.e4583.
3. Rahimikhanghah, A., Tajkey, M., Rezazadeh, B. and Rahmani, A.M., 2022. Resource scheduling methods in cloud and fog computing environments: a systematic literature review. Cluster Computing, pp.1-35.
4. Saxena, D. and Singh, A.K., 2024. A comprehensive survey on sustainable resource management in cloud computing environments. Authorea Preprints.
5. Singh, R.M., Awasthi, L.K. and Sikka, G., 2022. Towards metaheuristic scheduling techniques in cloud and fog: an extensive taxonomic review. ACM Computing Surveys (CSUR), 55(3), pp.1-43.
6. Wang, Z., Goudarzi, M., Gong, M. and Buyya, R., 2024. Deep reinforcement learning-based scheduling for optimizing system load and response time in edge and fog computing environments. Future Generation Computer Systems, 152, pp.55-69.
7. Subramoney, D. and Nyirenda, C.N., 2020, December. A comparative evaluation of population-based optimization algorithms for workflow scheduling in cloud-fog environments. In 2020 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 760-767). IEEE.
8. Anka, F., Tejani, G.G., Sharma, S.K. and Baljon, M., 2025. A Bioinspired Method for Optimal Task Scheduling in Fog-Cloud Environment. Computer Modeling in Engineering & Sciences (CMES), 142(3).
9. Ijaz, S., Ahmad, S.G., Ayyub, K., Munir, E.U. and Ramzan, N., 2025. Energy-efficient time and cost constraint scheduling algorithm using improved multi-objective differential evolution in fog computing. The Journal of Supercomputing, 81(1), p.116.
10. Azizi, S., Shojafar, M., Abawajy, J. and Buyya, R., 2022. Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach. Journal of network and computer applications, 201, p.103333.
11. Mastoi, Q.U.A., Ying Wah, T., Gopal Raj, R. and Lakhan, A., 2020. A novel cost-efficient framework for critical heartbeat task scheduling using the Internet of medical things in a fog cloud system. Sensors, 20(2), p.441.
12. Talaat, F.M., Ali, H.A., Saraya, M.S. and Saleh, A.I., 2022. Effective scheduling algorithm for load balancing in fog environment using CNN and MPSO. Knowledge and Information Systems, 64(3), pp.773-797.
13. Javanmardi, S., Shojafar, M., Persico, V. and Pescapè, A., 2021. FPFTS: A joint fuzzy particle swarm optimization mobility-aware approach to fog task scheduling algorithm for Internet of Things devices. Software: practice and experience, 51(12), pp.2519-2539.
14. Lin, C., Han, G., Qi, X., Guizani, M. and Shu, L., 2020. A distributed mobile fog computing scheme for mobile delay-sensitive applications in SDN-enabled vehicular networks. IEEE Transactions on Vehicular Technology, 69(5), pp.5481-5493.
15. Najafizadeh, A., Salajegheh, A., Rahmani, A.M. and Sahafi, A., 2022. Multi-objective Task Scheduling in cloud-fog computing using goal programming approach. Cluster Computing, 25(1), pp.141-165.
16. Madhura, R., Elizabeth, B.L. and Uthariaraj, V.R., 2021. An improved list-based task scheduling algorithm for fog computing environment. Computing, 103(7), pp.1353-1389.
17. Potu, N., Jatoth, C. and Parvataneni, P., 2021. Optimizing resource scheduling based on extended particle swarm optimization in fog computing environments. Concurrency and Computation: Practice and Experience, 33(23), p.e6163.
18. El-Nattat, A., Elkazzaz, S., El-Bahnasawy, N.A. and El-Sayed, A., 2021, July. Performance improvement of fog environment using deadline based scheduling algorithm. In 2021 International Conference on Electronic Engineering (ICEEM) (pp. 1-6). IEEE.
19. Dubey, K. and Sharma, S.C., 2021. A novel multi-objective CR-PSO task scheduling algorithm with deadline constraint in cloud computing. Sustainable Computing: Informatics and Systems, 32, p.100605.
20. Sun, Y., Lin, F. and Xu, H., 2018. Multi-objective optimization of resource scheduling in fog computing using an improved NSGA-II. Wireless Personal Communications, 102, pp.1369-1385.
21. Ahmad, M.A., Patra, S.S. and Barik, R.K., 2020. Energy-efficient resource scheduling in fog computing using SDN framework. In Progress in Computing, Analytics and Networking: Proceedings of ICCAN 2019 (pp. 567-578). Springer Singapore.