

An Effective analysis on various task scheduling algorithms in Fog computing

Prashanth Choppara¹, Sudheer Mangalampalli^{1,*}

¹School of Computer Science and Engineering, VIT-AP University, Amaravati, AP, India, 522237

Abstract

Fog computing involved as an extension of cloud and distributed systems fog nodes allowing data to be processed closer to the edge device and reduces the latency and bandwidth, storage capacity of IoT tasks. Task scheduling in fog computing involves allocating the tasks in fog nodes based on factors such as node availability, processing power, memory, and network connectivity. In task scheduling we have various scheduling algorithms that are nature inspired and bio-inspired algorithms but still we have latency issues because it is an NP-hard problem. This paper reviews the existing task scheduling algorithms modeled by metaheuristic, nature inspired and machine learning which address the various scheduling parameters like cost, response time, energy consumption, quality of services, execution time, resource utilization, makespan, throughput but still parameters like trust, fault tolerance not addressed by many of the existing authors. Trust and fault tolerance gives an impact and task scheduling trust is necessary to tasks and assign responsibility to systems, while fault tolerance ensures that the system can continue to operate even when failures occur. A balance of trust and fault tolerance gives a quality of service and efficient task scheduling therefore this paper done analysis on parameters like trust, fault tolerance and given research directions.

Keywords: Fog computing, Task scheduling, Machine learning

Received on 10 October 2023, accepted on 04 December 2023, published on 13 December 2023

Copyright © 2023 P. Choppara *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [CC BY-NC-SA 4.0](#), which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/eetiot.4589

1. Introduction

In this present era IoT devices are rapidly developing day by day according to international data corporation predicts (IDC) that by 2025, 55.8 billion devices to be connected worldwide 80% of them are connected to IoT devices [1]. Every IoT device generates a large amount of data. This data was processed in a cloud. The IoT devices are sending the data to the cloud distributed systems for storing, processing, analyzing, and decides due to its high computation tasks. Cloud computing offers numerous benefits, yet there are also some disadvantages like complexity, overhead, bottlenecks, resource allocation, Dependency on cloud provider and it can't satisfy the latency issue of IoT applications [2]. Because Fog computing wasn't a replacement of cloud it was to accompaniment cloud computing. Fog as the intermediate

layer of cloud and IoT applications in fog layer we have many fog nodes that are used to process the data before it is sent to the cloud. Fog computing provides the data processing, networking, analysis the data, so that cloud like services is close to the IoT application to reduce the latency period, energy competition, save bandwidth, better security for IoT applications [3]. System model of fog computing environment in between the cloud and IoT layer this a three-tier architecture (cloud-fog-IoT). The first layer includes all IoT devices that generated the data this data was pre-processed to fog and cloud layer. The second layer computed the resources such as edge servers, routers, gateways, switches, and cloudlets. The fog layer, cloud layer both are responsible for the processing, receiving data, and responding to IoT applications requests are send from the IoT application layer. The third layer consists of data centers, vms this cloud layer is connected to fog layer [4].

*Corresponding author. Email: sudheerkietmtech@gmail.com

Task scheduling is a process of allocating resources such as memory, CPU time, input and output processing to ensure that tasks are executed efficiently within a time manner that is mostly done by operating systems. It is an important component of the operating system and is used to optimize the system performance, improve resource utilization, and ensure that which task is executed first according to their tasks priority [5]. Task scheduling approaches in fog computing are centralized scheduling, decentralized scheduling, hybrid scheduling, adaptive scheduling. The centralized scheduling across the all-fog nodes is controlled by a central controller; this approach may lead to a single point of failure but can also improve resource allocation. The decentralized scheduling approach makes its own decision based on their local node information this is more reliable than centralized scheduling. Hybrid scheduling is a combination of centralized and decentralized approaches, allowing some scheduling decisions to be made local and while others are coordinated by a central controller. The adaptive scheduling approach uses machine learning to dynamically adjust the tasks scheduling decisions based on changing workload and resource availability. This approach can provide more efficient resource utilization but require more computational overhead [6].

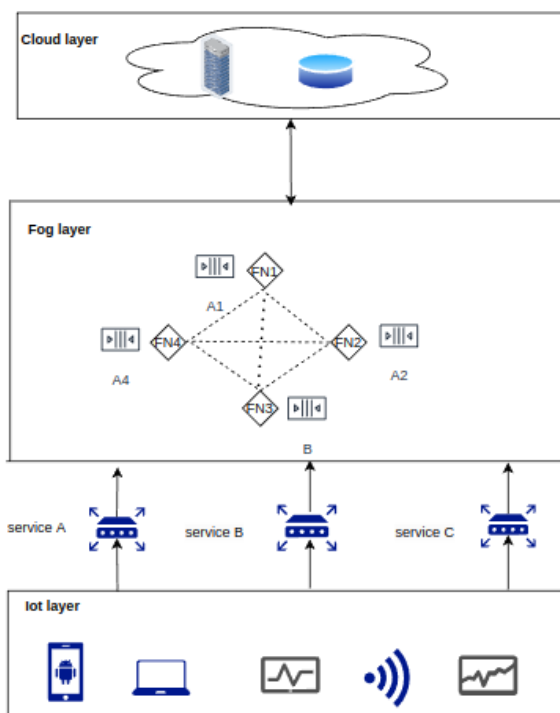


Figure 1. Fog computing environmental

Task scheduling is an NP hard problem. Many researchers believe there are no polynomial time algorithms for solving NP hard problems.

Contribution in this paper represents below:

1. Review of various existing task scheduling algorithms in fog computing
2. Efficient analysis done by existing task scheduling

3. After effective analysis on existing task scheduling algorithms few researchers' direction given and solve task scheduling problem in fog computing environments

In this section we discussed the importance of fog computing (fog nodes) and task scheduling. Rest of the paper is structured as below. Section 1 describes the Introduction, section 2 describes the Related Works, Section 3 describes research direction, section 4 describes conclusion.

2. Related works

In [7] the author formulated a task scheduling algorithm to tasks two approaches. QEESM reduce the energy consumption in a novel scheduling strategy that optimize the fog nodes. The proposed model is a statistical measure of moving average and Heiken-Ashi candlesticks patterns of the metric values obtained from a set of make spans each fog node sequences. It was simulated on the ifogsim library. It compared existing QEESM approach and from result it proves that minimized the energy consumption. Parameters concentrated on computation load, roundtrip time and request receiving time. In [8] authors formulated the scheduling to minimize the total energy consumption of the fog layer. It is developed by the two semi-greedy based algorithms multi start procedure (PSG-M) and priority-aware semi-greedy (PSG). It is simulated in C++ visual studio. The proposed algorithms compare against the First Come First Serve, Greedy for Energy (GE), Earliest Deadline First, Detour researcher achieve minimize the energy consumed and response time by fog nodes. In [9] the author designed AEOSSA is a task scheduling for IoT applications requests in a cloud-fog environment that is (AEO) artificial ecosystem-based optimization. This version is formulated utilizing salp swarm algorithm operators to harness the ability of AEO throughout the processing of finding the optimal solution to the problem. This simulation carried out the MATLAB R2018b and different datasets of different sizes to tackle the task scheduling in AEOSSA approach. It is evaluated against the existing Harris Hawks Optimization, Particle Swarm Optimization, Salp Swarm Algorithms and Firefly Algorithm. The result demonstrated the high ability to AEOSSA perform better than other methods the proposed method to optimize the task execution time, make span, and throughput. In [10] the author developed a Bag-of-Tasks workload model study the possibilities of using every resource is available in order to reduce the costs. It is simulated by the java parallel processing framework (JPPF). The existing model weighted Round Robin Algorithm simulated results are shown that combines every available resource can reduce the costs, meanwhile mean response time is not increasing. In [11] author formulated two type of approaches first method is using the (DVFS) Dynamic Voltage and Frequency Scaling technique to minimizing the energy consumption and second method sequence to construct the valid task a hybrid (IWO-CA) invasive weed optimization and culture evolutionary algorithm is applied. It

is simulated by C# programming. It compared against the HEFT_B, HEFT-T and DEWTS approach and observe the results showed that it is dominant over existing algorithm approaches for task scheduling issues to reduce energy consumption and minimizing the makespan. In [12] the author designs two approaches in energy aware task scheduling delay mechanisms: mixed-integer nonlinear programming (MINLP) and fuzzy-based reinforcement learning. MINLP to minimize the energy consumption, fuzzy-based deep reinforcement learning is to minimize the service delay of the tasks and energy usage of the fog computing. It is simulated by python language in Jupiter notebook. It evaluated against the First Come First Serve, Greedy for Energy, Detour, Earliest Deadline First, Priority aware semi greedy approach the result shown significant impact over existing baseline approach minimizes the average service time and energy consumption of the fog node. In [13] the author developed a Hunter Plus approach which examined the Gated Recurrent Unit to a Bidirectional Gated Recurrent Unit GGCN's to reduce the energy per task and job competed. The datasets were generated using a random scheduler data for hundred intervals for two different data workloads. It simulates the pyfog python library. It evaluated over baseline FCFS, Max-min, MESF, GA, ACO. The results or revealed GGCN's shows the large impact over existing algorithms for parameters i.e., job completion and energy consumption. In [14] the author formulated Iot tasks a DRL based algorithms named Clipped Double Deep Q- learning using target network experience replays techniques to minimize the energy, cost, and service delay. It is a python based discrete event simulation framework SimPy. It evaluated against random scheduling, First come first serve, random scheduling and q-learning scheduling simulation results revealed that Clipped Double Deep Q-learning outperforms compared approaches by minimizing the cost, service delay, energy consumption. In [15] the author developed a novel efficient reinforcement algorithm's which combines optimization with stable lower bounds. The batch of tasks at a time decide on task selection through machine learning algorithms. It simulated the python framework SimPy and the dataset contains 600 tasks that arrive in 1000 seconds. The results are compared against the baseline algorithm MIN-MIN, (RDA) random algorithm shows to minimize the task runtime. In [16] authors designed a task scheduling algorithm which addresses parameters load balancing and average delay the different numbers of Iot tasks. MDP (Markov decision process) used as a methodology to gear task scheduling problems. All simulation work conducted on jupyter notebook using python library fogpy. This strategy uses randomized worklogs as workloads. Based on simulation results, the author's proposed technique demonstrated a considerable improvement over existing parameters in terms of average end-to-end delay for schemes with and without IoT task and load balancing considerations. In [17] Based on the pareto optimality, a task scheduling algorithm is designed with the goal of achieving a trade-off between the make span and

carbon emission reduction. It modeled Marine Predator's algorithms with the Polynomial Mutation mechanism. It simulated the Jmetal framework. It compares over SMPO, OMOPSO, DMOPSO. Results proved that Marine predator's algorithm with the polynomial mutation mechanism showed huge impact by improvising parameters makes span, energy consumption. In [18] the author developed three phases' algorithms: first level phase independent tasks are identified, second phase assigns priority to the task and third phase balanced combination of global optimal and optimal approach. This approach designs the DAG algorithm to minimize the overall computation and cost, makespan of the processors. It simulated in the iFogSim framework written in the Java programming language. To evaluate performance with PEFT, HEFT, MOPT shows good results for all existing algorithms. In [19] the author formulated the task scheduling (ILP) Integer linear programming optimization model concentrated on both time and energy consumption. Researchers designed (OppoCWOA) opposition-based chaotic whale optimization algorithm. Its simulation in python with PyCharm editor. The evaluated against the existing meta-heuristic algorithms Particle Swarm Optimization, Genetic algorithm, Artificial Bee Colony and result proved that OppoCWOA outperforms baseline algorithm by minimizing the energy consumption and time. In [20] the author designed a low latency task scheduling approach based on meta-heuristic scheduler Smart Ant Colony Optimization (SACO). It was simulated on MATLAB 2014a. It was evaluated against the modified particle swarm optimization (MPSO), Round Robin (RR), throttled scheduler algorithm, and Bee life algorithm (BLA). From the result it is evident that SACO showed significant impact over existing baseline approaches for specific metrics. In [21] authors focused on energy consumption and reducing delay as they developed a Multi-Objectives Grey Wolf Optimizer algorithm held in the fog broker to analyzed, estimate, and then schedule all sending requests from devices. Entire simulation carried out on MATLAB R2018b. It compares to existing NSG, MOPSO approaches. Results it proved that, reducing delay and energy consumption. In [22] author model formulated using hybrid approach fireworks algorithm (FWA), Earliest Finish Time (HEFT). The bi-objective optimization approach presented to reduce the cost factors, makespan. It is implemented in the ifogsim toolkit developed by the java programming language. The proposed BH-FWA are analyses and compared with the IMFWA, HEFT. From results it showed that 12% of improves in makespan. In [23] author developed a task scheduling algorithm called electric earthworm optimization algorithm for this algorithm using real-world workloads HPC2N and CEA-CURIE. It simulated on cloud Sim. It evaluated against the existing algorithm cuckoo search cuckoo search and differential evolution algorithm, particle swarm optimization (CSPO), hybrid oppositional differential evolution-enabled whale optimization algorithm results are shown minimizing the makespan, execution time, cost, and energy consumption. In [24] the author developed a dynamic priority for workflow

scheduling using multi-objective normalization workflow scheduling. The entire simulation was done in cloud Sim, the results are compared existing algorithms DLS, HEFT, max-min, min-min proposed to improve in makespan, increases in maximum average cloud utilization and decrease in cost. In [25] the author designed an efficient task scheduling algorithm trust-aware scheduling algorithm using firefly optimization. For this approaches workload using fabricated

datasets with different distributions real time worklogs of NASA and HPC2N. It was simulated in cloud Simpy environment. Results are compares over the baseline approach. GA, ACO and PSO showing a huge impact over reduces the makespan, success rate, availability, and turn-around efficiency

Table 1. parameters addressed by varied existing algorithms

Authors	Techniques	Simulation tool	Parameters
[7]	Statistical measures moving average and Heiken-Ashi patterns Quality-aware energy efficient scheduling model	iFogsim	Makespan, energy consumption
[8]	PSG with multi start procedure (PSG-M) & priority-aware semi-greedy (PSG)	C++ visual studio	Response time, energy consumption
[9]	AEOSSA (artificial ecosystem-based optimization salp swarm algorithm)	MATLAB R2018b	Makespan, task execution time and throughput
[10]	Bag-of-tasks workload model	JPPF framework	Cost, response time
[11]	DVFS (Dynamic voltage and frequency scaling)	C#	Energy consumption, makespan
[12]	Mixed integer nonlinear programming (MINLP) and Fuzzy-based reinforcement learning	Python	Average service time and energy consumption
[13]	GCNN'S	PyFog	Energy consumption, job completion
[14]	Clipped Double Deep Q-learning using target network	Simpy	Cost, service delay, energy consumption
[15]	Reinforcement learning algorithm	Simpy	Task runtime
[16]	Markov decision process	Fogpy	Average end-to-end delay, load balancing
[17]	Marine Predator's Algorithm with the Polynomial Mutation Mechanism in fog computing	Jmetal framework	Makespan, energy consumption
[18]	(DAG) Directed acyclic graphs	IFogSim	Makespan, cost
[19]	Opposition-based Chaotic Whale Optimization Algorithm (OppoCWOA)	Python pycharm editor	Time and energy consumption
[20]	Smart Ant Colony Optimization (SACO)	MATLAB 2014a	latency

[21]	multi-objectives Grey Wolf Optimizer algorithm (MGWO)	MATLAB R2018b	Delay and energy consumption
[22]	Bi-objective optimization	iFogSim	Makespan and cost
[23]	Electric earthworm optimization algorithm (EEOA)	Cloud Sim	Execution time, cost, Makespan
[24]	MONWS (multi-objective normalization workflow scheduling)	Cloud Sim	Makespan and cost
[25]	(TAFFA) Trust-Aware scheduling algorithm using firefly optimization	cloud Sim	Makespan, turnaround efficiency and success

From the above Table:1 existing algorithms and authors are focused on nature inspired, bio inspired and metaheuristic algorithms they achieved a lot but still we have to optimize in task scheduling because it is a NP-hard problem. Researches are focused on machine learning based techniques in fog computing deep reinforcement learning algorithms like Q-learning, MDP, queue learning, stochastic process etc., more scope of performance in task scheduling

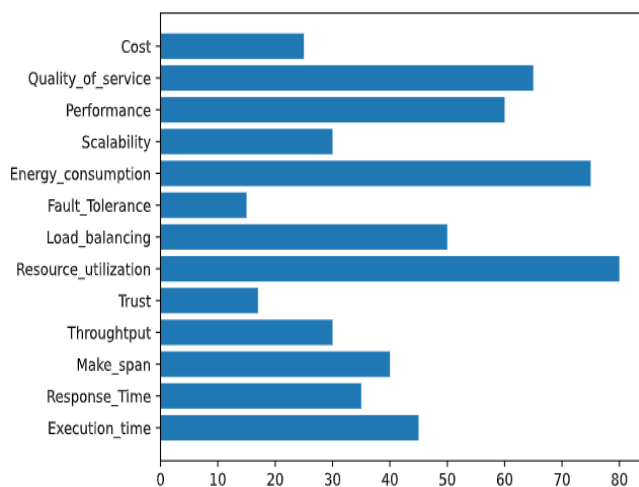


Figure 2. Parameters addressed various task scheduling algorithms

3. Research directions

Based on the literature we address, most of the scheduling algorithms are focused on quality of service, performance, scalability, energy consumption, load balancing, resource utilization, response time, and make span. While observed the fig2 trust and fault tolerance have a scope to improve. A machine learning can optimize the task scheduling process and improve the overall efficiency of the fog nodes.

3.1. Trust aware scheduling

Trust aware scheduling is necessary in fog computing environment because it supports to address privacy and security in the distributed computing environment. With fog computing, the devices close to the network edge, which means that they are more vulnerable to security threats. Trust aware scheduling helps to overcome these risks by allocating resources to trusted components of the system. In fog computing trust determines various components, such as the security mechanism in place, the reputation of the component, and the level of access it has to sensitive data. Trust-aware scheduling helps better the overall execution in fog computing systems allocating resources reduce risk of resource wastage.

3.2. Fault tolerance scheduling

It is an important aspect to check the capability of the system. In fog computing ensures reliable and efficient task execution. Task scheduling in fog computing has different approaches. One is redundancy, where multi copies of task are executed by several nodes to ensure that at least one copy is completely successful. Another approach was checkpointing, which involves periodically saving the progress of a task to persistent storage, allowing for the resumption of the task from the last checkpoint if a node fails. To detect and handle the failures' fault detection and recovery mechanism to be implemented. For instance, monitoring tools can detect when a node becomes unresponsive, and the automotive recovery process can be transferred to another available fog node.

4. Conclusion

Task scheduling is an important aspect in cloud and fog computing because it is played a particular role in the optimizing use of computing assets. Fog computing helps to reduce overhead data transport and, as a result, increases computational efficiency in cloud networks by eliminating the need to analyses and maintain large amount of data. In this paper review, recently modeled task scheduling algorithms both cloud computing and fog computing environments have been studied and various task scheduling parameters are used to compare existing algorithms. priority based scheduling, bag of tasks workload model, DAG algorithm, DVFS, salp swarm algorithm, whale optimization algorithm, electric earthworm optimization algorithms based on nature inspired bio-inspired and meta-heuristic. From fig.2 we can see that resource utilization and system performance are covered in 70 to 80 percent of discussed algorithms for improvement. But still there is a lot of work required to enhance the trust and fault tolerance. An efficient task scheduling fulfills all requirements of the cloud-fog paradigm. While we are using deep reinforcement learning in fog computing task scheduling to optimize the better resource allocation and task execution. By using a DRL agent to learn the optimal policy for allocating tasks to fog nodes based on the available resource and task requirements.

Conflict of interest

Authors declare that there are no conflicts of interest.

Acknowledgements

I would like to express my sincere gratitude to my Ph.D. research supervisor Dr. Sudheer Mangalampalli for his valuable and constructive suggestions during the planning and development of this research work. Financially, this work is not supported by any organisation/funding agency.

References

- [1] Kishor, A., & Chakarbarty, C. (2021). Task offloading in fog computing for using smart ant colony optimization. *Wireless personal communications*, 1-22
- [2] Ramezani Shahidani, F., Ghasemi, A., Toroghi Haghghat, A., & Keshavarzi, A. (2023). Task scheduling in edge-fog-cloud architecture: a multi-objective load balancing approach using reinforcement learning algorithm. *Computing*, 105(6), 1337-1359.
- [3] Tyagi, R., & Gupta, S. K. (2018). A survey on scheduling algorithms for parallel and distributed systems. In *Silicon Photonics & High Performance Computing: Proceedings of CSI 2015* (pp. 51-64). Springer Singapore.
- [4] Hamid, S. A. (2023). Fog Computing Architecture in higher education institutions. *Eurasian Research Bulletin*, 17, 92-99.
- [5] Dai, Z., Ding, W., Min, Q., Gu, C., Yao, B., & Shen, X. (2023). ME-AWA: A Novel Task Scheduling Approach Based on Weight Vector Adaptive Updating for Fog Computing. *Processes*, 11(4), 1053.
- [6] Baek, J. Y., Kaddoum, G., Garg, S., Kaur, K., & Gravel, V. (2019, April). Managing fog networks using reinforcement learning based load balancing algorithm. In *2019 IEEE Wireless Communications and Networking Conference (WCNC)* (pp. 1-7). IEEE.
- [7] Potu, N., Bhukya, S., Jatoth, C., & Parvataneni, P. (2022). Quality-aware energy efficient scheduling model for fog computing comprised IoT network. *Computers & Electrical Engineering*, 97, 107603.
- [8] Azizi, S., Shojafar, M., Abawajy, J., & 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, 103333.
- [9] Abd Elaziz, M., Abualigah, L., & Attiya, I. (2021). Advanced optimization technique for scheduling IoT tasks in cloud-fog computing environments. *Future Generation Computer Systems*, 124, 142-154.
- [10] Tychalas, D., & Karatza, H. (2020). A scheduling algorithm for a fog computing system with bag-of-tasks jobs: Simulation and performance evaluation. *Simulation Modelling Practice and Theory*, 98, 101982.
- [11] Hosseinioun, P., Kheirabadi, M., Tabbakh, S. R. K., & Ghaemi, R. (2020). A new energy-aware tasks scheduling approach in fog computing using hybrid meta-heuristic algorithm. *Journal of Parallel and Distributed Computing*, 143, 88-96.
- [12] Raju, M. R., & Mothku, S. K. (2023). Delay and energy aware task scheduling mechanism for fog-enabled IoT applications: A reinforcement learning approach. *Computer Networks*, 224, 109603.
- [13] Iftikhar, S., Ahmad, M. M. M., Tuli, S., Chowdhury, D., Xu, M., Gill, S. S., & Uhlig, S. (2023). HunterPlus: AI based energy-efficient task scheduling for cloud-fog computing environments. *Internet of Things*, 21, 100667.
- [14] Swarup, S., Shakshuki, E. M., & Yasar, A. (2021). Energy efficient task scheduling in fog environment using deep reinforcement learning approach. *Procedia Computer Science*, 191, 65-75.
- [15] Jin, C., Han, Y., Deng, Z., Chen, Y., Liu, C., & Huang, J. (2023). Reinforcement Learning-Based Intelligent Task Scheduling for Large-Scale IoT Systems. *Wireless Communications and Mobile Computing*, 2023.
- [16] Razaq, M. M., Rahim, S., Tak, B., & Peng, L. (2022). Fragmented task scheduling for load-balanced fog computing based on Q-learning. *Wireless Communications and Mobile Computing*, 2022.
- [17] Abdel-Basset, M., Moustafa, N., Mohamed, R., Elkomy, O. M., & Abouhawwash, M. (2021). Multi-objective task scheduling approach for fog computing. *IEEE Access*, 9, 126988-127009.
- [18] Madhura, R., Elizabeth, B. L., & Uthariaraj, V. R. (2021). An improved list-based task scheduling algorithm for fog computing environment. *Computing*, 103, 1353-1389.
- [19] Movahedi, Z., & Defude, B. (2021). An efficient population-based multi-objective task scheduling approach in fog computing systems. *Journal of Cloud Computing*, 10(1), 1-31.
- [20] Kishor, A., & Chakarbarty, C. (2021). Task offloading in fog computing for using smart ant colony optimization. *Wireless personal communications*, 1-22.
- [21] Saif, F. A., Latip, R., Hanapi, Z. M., & Shafinah, K. (2023). Multi-objective grey wolf optimizer algorithm for task

- scheduling in cloud-fog computing. *IEEE Access*, 11, 20635-20646.
- [22] Yadav, A. M., Tripathi, K. N., & Sharma, S. C. (2022). A bi-objective task scheduling approach in fog computing using hybrid fireworks algorithm. *The Journal of Supercomputing*, 78(3), 4236-4260..
- [23] Kumar, M. S., & Karri, G. R. (2023). Eeoa: cost and energy efficient task scheduling in a cloud-fog framework. *Sensors*, 23(5), 2445.
- [24] Pillareddy, V. R., & Karri, G. R. (2023). MONWS: Multi-Objective Normalization Workflow Scheduling for Cloud Computing. *Applied Sciences*, 13(2), 1101.
- [25] Mangalampalli, S., Karri, G. R., & Elngar, A. A. (2023). An Efficient Trust-Aware Task Scheduling Algorithm in Cloud Computing Using Firefly Optimization. *Sensors*, 23(3), 1384.