Effective task allocation in fog computing environments using fractional selectivity model

Prasanna Kumar Kannughatta Ranganna[1], Siddesh Gaddadevara Matt[2], Ananda Babu Jayachandra[3], Vasantha Kumara Mahadevachar[4]

[1]Department of Computer Science, M. S. Ramaiah Institute of Technology, Visvesvaraya Technological University, Belagavi, India
[2]Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), M. S. Ramaiah Institute of Technology, Visvesvaraya Technological University, Belagavi, India
[3]Department of Information Science, Malnad College of Engineering, Visvesvaraya Technological University, Belagavi, India
[4]Department of Computer Science, Government Engineering College, Visvesvaraya Technological University, Belagavi, India

Article Info

Article history:

Received Jul 29, 2024
Revised Nov 29, 2024
Accepted Jan 27, 2025

Keywords:

Fog computing
Fractional selectivity
Internet of things
Resource utilization
Task allocation

ABSTRACT

In recent scenario, fog computing is a new technology deployed between cloud computing systems and internet of things (IoT) devices to filter out important information from a massive amount of collected IoT data. Cloud computing offers several advantages, but also has the disadvantages of high latency and network congestion, when processing a vast amount of data collected from various devices and sources. For overcoming these problems in fog computing environments, an efficient model is proposed in this article for precise load balancing (LB). The proposed fractional selectivity model significantly handles LB in fog computing by reducing network bandwidth consumption, latency, task-waiting time, and also enhances the quality of experience. The proposed model allocates the required resources by eliminating sleepy, unreferenced, and long-time inactive services. The fractional selectivity model's performance is investigated on three application scenarios, namely virtual reality (VR) game, electroencephalogram (EEG) healthcare, and toy game. The efficiency of the introduced model is analyzed on the basis of makespan, average resource utilization (ARU), load balancing level (LBL), total cost, delay, and energy consumption. Specifically, in comparison to the traditional task allocation models, the proposed model reduces almost 5 to 15% of the total cost and makespan time.

Corresponding Author:

Prasanna Kumar Kannughatta Ranganna
Research Scholar, Department of Computer Science, M. S. Ramaiah Institute of Technology
Visvesvaraya Technological University
Belagavi, India
Email: prasanna.kghatta@sit.ac.in

1. INTRODUCTION

Currently, internet of things (IoT) technology facilitates internet-connected devices for communicating with each other in order to achieve common objectives [1]. In the present decade, approximately 30 billion IoT devices are in operation, and by the year 2025, it is expected to reach 80 billion. The use of IoT devices is increasing dramatically, leading to the generation of a vast amount of heterogeneous data [2]. Recently, IoT devices are extensively applied in several applications, namely smart agriculture, traffic monitoring, health, smart homes, and animal tracking [3]–[5]. However, most of the IoT devices have limited storage capacity and processing power. These IoT devices are incompatible with extensive computational applications because they consume more energy [6]. As a solution, cloud-computing paradigms are utilized for executing IoT applications [7]. In a few circumstances, IoT devices also suffer from the problems of delay and poor bandwidth while interacting with the cloud servers.

In addition to this, the big data generated from the IoT devices also leads to cloud server's congestion. The cloud data centers are dense, often causing high delays and network congestion for outlying requests [8], [9]. Fog computing is an emerging technology used to overcome the above-stated constraints, to meet the requirements of IoT-based applications [10], [11]. In fog computing systems, load balancing (LB) is crucial in order to avoid latency. LB is the process of distributing tasks or requests in computing environments that guarantees the reliability and throughput [12]. It is generally difficult to control the execution service of the requests when the number of user requests increases. The poor control of computing systems causes more power consumption and lower throughput [13], [14].

Therefore, LB is a crucial aspect in maintaining business continuity in both distributed and parallel computing environments [15]. In this article, a novel model is proposed for effective distribution of user tasks or requests on various computing resources, with a high degree of task allocation and LB. The proposed fractional selectivity model provides a substantial solution to manage LB in fog computing, which produces better network performance. Especially, this method minimizes the network energy consumption, latency and makespan time, thereby improving user's overall quality. A special feature of this model is having the capability to optimize the resources effectively by removing irrelevant services which are inactive for long time periods. For an effective evaluation, numerous state-of-the-art models such as genetic algorithm (GA), particle swarm optimization (PSO), non-dominated sorting genetic algorithm II (NSGA-II), Bees, and interior point method (IPM) are exploited for showing the effectiveness of proposed model. The contributions of this article are outlined as follows:

– Proposed fractional selectivity model for task allocation in fog computing systems. The proposed model allocates a fraction of incoming tasks or data to every fog node, based on the requirements and properties. This model effectively optimizes resource usage that ensures better storage capacity and appropriate processing power for the fog nodes.
– Task allocation based on fractional selectivity is dynamic in real-time application scenarios. The fog nodes continuously monitor the workload and adjust the fraction of tasks in order to maintain optimal performance. The fractional selectivity enables significant task distribution that decreases system cost and response time in fog computing architectures.
– Conducted a series of experiments by varying the number of tasks, utilizing iFogSim toolkit, for evaluating the efficiency and effectiveness of the fractional selectivity model by means of makespan, average resource utilization (ARU), load balancing level (LBL), total cost, delay, and energy consumption.

This article is structured as follows. Literature review of existing models on the topic of "task allocation in fog computing systems" is presented in section 2. The theoretical explanation, numerical analysis, and conclusion of the proposed fractional selectivity model are specified in sections 3, 4, and 5 respectively.

2. LITERATURE REVIEW

Kaur and Aron [16] introduced a hybrid model (water cycle optimization algorithm, simulated annealing, and plant growth optimization algorithm) for executing workflow tasks in fog computing by efficiently balancing the load. Additionally, a fog-clustering algorithm was developed in this study for reducing execution time, computational cost, and energy consumption, while executing the tasks related to workflow in fog-cloud environments. The developed hybrid model was simulated using the iFogSim toolkit, and its effectiveness was validated in light of cost, energy consumption, and time delay. Similarly, Gupta and Singh [17] presented a dynamic LB model in fog-IoT environments by hybridizing two metaheuristic algorithms, namely grey wolf optimization (GWO) algorithm and modified Moth-flame optimization algorithm. However, running multiple optimization algorithms introduced performance overhead, along with increasing the overall complexity of the model.

Talaat *et al.* [18] combined a modified PSO algorithm with convolutional neural network (CNN) for dynamic LB in fog-computing environments. In comparison to other LB models, the presented model significantly decreased the response time with better resource usage. Empirical outcomes stated that the presented LB model was efficient and simple in real-time fog computing systems, particularly related to healthcare applications. The presented model obtained better LBL, ARU, and makespan related to traditional LB models. However, task failures occurred with the presented model due to heavy demand on the servers hosting the workflow tasks.

Talaat *et al.* [19] introduced a simple and dynamic LB model by integrating GAs and reinforcement learning. This LB model continuously monitored traffic in fog computing systems, acquired load information from every server, managed requests, and precisely distributed the load among the servers. The presented LB model enhanced quality of services (QoS) in fog-cloud computing environments by means of response time and cost allocation. Additionally, it ensured continuous service by efficiently establishing resource utilization. Yet, the presented LB model caused bottleneck problems by continuously monitoring traffic in fog computing systems.

Kaur and Aron [20] implemented a hybridized LB model to enhance resource utilization and reduce latency in fog computing applications. This hybridized LB model incorporated three algorithms, namely ant colony optimization (ACO) algorithm, tabu search, and GWO algorithm. In this study, the presented hybridized LB model was simulated using the Eclipse and iFogSim toolkits. Similarly, Hussein and Mousa [21] integrated two metaheuristic optimization algorithms, namely PSO and ACO to balance the load in fog computing systems with minimal response time and communication costs. However, the performance overhead and complexity were the two major problems while hybridizing more optimization algorithms in fog computing systems.

Singh *et al.* [22] developed a LB model for enhancing resource utilization in software defined network (SDN) enabled fog environments. Additionally, a deep belief network (DBN) was employed for intrusion detection that decreased communication delays in the fog layer. The results stated that the presented model significantly reduced communication delays, average energy consumption, and average response time, better than the conventional models. Furthermore, Yakubu and Murali [23] initially used a layer fit strategy for distributing tasks between the cloud and fog, based on priority levels. Then, a modified Harris hawks' optimization (HHO) algorithm was designed for effective task scheduling. The primary objective of this study was to improve resource usage and reduce power consumption, task execution cost, and makespan time, in both the cloud and fog layers.

Baburao *et al.* [24] introduced an efficient dynamic resource allocation model based on PSO algorithm for handling the LB problems in fog computing. The presented model significantly allocated the required resources by eliminating sleepy, unreferenced and long-time inactive services from random access memory. Javaheri *et al.* [25] initially developed a hidden Markov model (HMM) based on Viterbi and Baum-Welch algorithms to predict the availability of every fog-computing provider by considering the factors like offload tasks, incoming requests, and deadline-missed workflows. Further, a discrete opposition based HHO algorithm was introduced for precise workflow scheduling. Still, the DBN, HHO and PSO algorithms faced challenges in adapting to rapidly changing environments in fog computing. Kishor and Chakarbarty [26] introduced a smart ACO algorithm to offload the tasks of IoT applications in fog computing environments. However, this study utilized only single-point connections between fog and cloud, and employed only a single-user system.

In addition, Singh [27] developed a novel LB model for fog computing by integrating a fuzzy algorithm with the golden eagle optimization algorithm (GEOA). The presented LB model encompassed of three phases, namely task prioritization, ranking and scheduling of resources, and power management. Firstly, a fuzzy algorithm was employed for assigning priorities to incoming tasks based on predefined priority, task size, and deadline time. By using a fuzzy algorithm, the task prioritization executed important tasks without any delay. Secondly, GEOA was applied for ranking and scheduling resources that ensured that the tasks were allocated to appropriate resources for efficient execution. Finally, a power management engine was implemented to optimize power consumption by disabling and enabling resources based on the necessity. Six different evaluation measures, namely waiting time, average turnaround time, communication overhead, computational cost, failure rate, and energy consumption were used for assessing the efficacy of the model. Nonetheless, running resource intensive optimization algorithms like GEOA on IoT devices, generally led to resource contention.

Natesha and Guddeti [28] introduced an elitism based genetic algorithm (EGA) to solve multi-objective problems in fog computing environments. The EGA ensured QoS requirements of IoT applications and minimized cost, energy consumption, and service time. The empirical evaluation indicated that the EGA outperformed existing algorithms in terms of service time, energy consumption, and service cost. The primary concern of this study was identifying appropriate fog devices (nodes) which were distributed and varied by means of service time, response time, data processing speed, resource availability. These fog nodes were utilized to process the data and host IoT applications. Also, Bey *et al.* [29] developed a quantum computing inspired model based on a neural network, for task allocation in IoT-edge computing environments. The developed model efficiently predicted optimal computing nodes in order to deliver real-time services. However, the developed fog-computing model was ineffective in managing the increased data volume and processing requirements. For addressing the aforementioned concerns, a novel task allocation model named fractional selectivity is proposed in this article. The advantages and disadvantages of existing studies are illustrated in Table 1.

Table 1. Advantages and disadvantages of existing studies

Author	Advantages	Disadvantages
Kaur and Aron [16]	Limited energy consumption	Priority is not considered in the distribution of tasks
Gupta and Singh [17]	Limited response time and loss rate	Increased total cost
Talaat *et al.* [18]	Reduced delay	Consumed lot of energy
Talaat *et al.* [19]	Reduced total cost	Task scheduling and LB consumed a lot of time
Kaur and Aron [20]	Decreased makespan	Increased response time
Hussein and Mousa [21]	Minimized bandwidth cost and resources	High computational cost
Singh *et al.* [22]	Reduced ARU and makespan	The allocation of resources does not consider the current utilization of fog nodes
Yakubu and Murali [23]	Reduction in balanced network and network delay	High failure rate
Baburao *et al.* [24]	Decreased energy consumption	Has high makespan and delay
Javaheri *et al.* [25]	Limited response time and loss rate	High power consumption
Kishor and Chakarbarty [26]	Decreased delay and bandwidth cost	Task priority is not considered
Singh [27]	Decreased energy consumption	Increased response time
Natesha and Guddeti [28]	Decreased makespan and delay	Increased energy consumption
Bey *et al.* [29]	Decreased total cost	Increased energy consumption

3. METHOD

The proposed fog computing system includes three layers, namely sensor layer, fog layer, and cloud layer. The sensor layer is also called device or edge layer, which is the lowest tier in distributed computing environments/architectures [30]. The sensor layer comprises of numerous sensors and physical devices which collect data from different aspects of the physical environment. The sensors include global positioning system (GPS) devices, motion detectors, cameras, and temperature sensors. The primary objective of the sensor layer is to collect data from the monitoring systems and application scenarios. The collected data is related to several parameters such as, user interactions, machine performance, and environmental conditions [31]. The sensor layer has limited storage capacity and processing capability. It collects data at predetermined intervals and then transmits the respective data to higher layers for further analysis and processing.

Correspondingly, the fog layer is also called edge-computing layer which is an intermediate layer between the cloud layer and the sensor layer in distributed computing environments/architectures. The term ‘fog’ denotes a computing environment, which is closer to the sensors that are related to the ‘remote data centers’ [32]. In this layer, the data collected from the sensor layer is analyzed and processed locally in near-real-time and real-time scenarios. This process helps in faster decision making with reduced latency. In this layer, fog computing includes gateways, edge servers, and computing resources. These devices run algorithms and applications for preprocessing, aggregating, and filtering data before transmitting it to the cloud layer. Fog computing is especially crucial in applications developed for smart cities, autonomous vehicles, and industrial automation, because it provides better data privacy, bandwidth optimization, and lower latency [33].

Finally, the cloud layer is a cloud-computing layer, which is the topmost layer in distributed computing environments. This layer comprises remote data centers which offer more storage, services, and computing resources over the internet. The data acquired from the sensors and further processed in the fog layer, is then analyzed, managed, stored, and used in the cloud layer [34]. Cloud computing provides centralized management, redundancy, and scalability for application scenarios. The cloud services include artificial intelligence, machine learning, data analytics, and advanced computing. The organization accesses cloud resources, and makes it a cost-effective and flexible solution for several applications such as data storage, e-commerce, and web services.

3.1. Research objectives

The main motivation behind this research is to handle the resources in the fog nodes to reduce complexity and efficiently accomplish the tasks. By optimally allocating the tasks in cloud-fog-IoT device environments, this work produces a higher-level efficiency in fog resource management. The main objectives that are achieved in this article are outlined as follows:

- As per the task requirements, the availability of the resources are checked and further, the resources are listed.
- A novel task allocation model is proposed in fog computing environments for decreasing the network usage and response time.
- Based on the energy consumption and resource availability, the tasks are allocated in order to efficiently minimize the task execution time. The detailed explanation about the proposed fractional selectivity

based task allocation model is presented in sections 3.1 and 3.2. The block-diagram of the proposed model is mentioned in Figure 1.

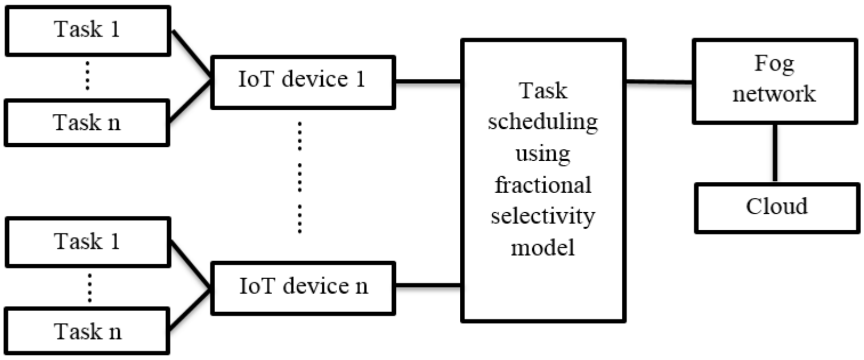

Figure 1. Block-diagram of the proposed model

3.2. Fractional selectivity model

The primary objective of the proposed fractional selectivity model is to optimize resource allocation among fog nodes. This model aims at reducing response times, enhancing resource utilization, and improving overall cost effectiveness. In the context of iFogSim, fractional selectivity refers to a concept or mechanism utilized for allocating computing resources in fog computing systems based on the fraction of tasks or data that needs to be processed at several fog nodes. iFogSim is one of the effective simulation frameworks used to simulate the proposed model in fog-computing environments. In this scenario, data is typically distributed across several fog nodes, which are positioned close to edge devices for improving the efficiency and reducing latency.

In a fog computing system, the introduced model is employed for optimizing resource allocation and data processing in distributed computing environments. In the present scenario, fog computing extends its abilities in cloud computing and is made of numerous sensors and devices in IoT ecosystems. In this context, the fractional selectivity model plays a critical role in enhancing the effectiveness and efficiency of data processing in a fog computing system. The fog nodes are considered as the computing resources which often have limited energy resources, memory, and computational power related to cloud servers. The fractional selectivity model assists in filtering out irrelevant data, and processes the necessary information, thus reducing the energy consumption and resource utilization.

The fractional selectivity makes decision-making faster in applications like control systems and real-time monitoring systems; here, lower latency is crucial. Fog computing responds quickly to the events by filtering and processing relevant data at the edges, and this process superiorly reduces the delay between the action and data generation. The transmission of a huge amount of data to the cloud is costly by means of bandwidth, and also leads to network congestion. The fractional selectivity reduces the size of data which needs to be transmitted to the cloud by filtering and data pre-processing. It ensures that only necessary data is transmitted to remote servers and conserves bandwidth. In a few circumstances, some data is confidential and sensitive which should not be sent over the network to the cloud. In this scenario, this model performs local data processing that ensures that the sensitive information remains within the controlled edge environments by improving security and privacy.

The cloud-fog computing environment has a vast amount of sensors and devices. The fractional selectivity-based task allocation model, efficiently scales the fog computing systems by distributing the load amongst fog nodes, and makes an optimal usage of available resources. The fractional selectivity leads to cost savings in terms of computing resources and cloud storage, through decreasing the amount of data processed and transmitted in the cloud. Overall, in fog computing systems, the fractional selectivity is a valuable model in achieving cost savings, enhancing security and privacy, conserving bandwidth, decreasing latency, and enhancing resource efficiency. The fractional selectivity enables all fog nodes to make an intelligent decision about data that is processed locally and transmitted to the cloud. This action provides more efficient and responsive edge computing solutions. The working process of the fractional selectivity model is denoted in Figure 2.

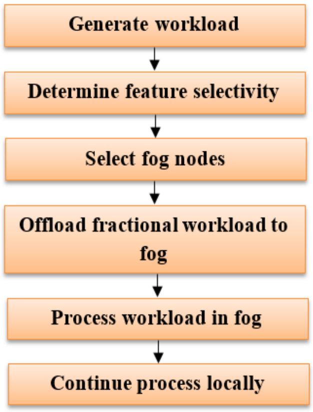

Figure 2. Working process of the fractional selectivity model

3.3. Innovativeness of fractional selectivity in fog computing

The fractional selectivity-based task allocation model is a flexible and fine-grained model in resource scheduling as it significantly optimizes resource allocation in distributed computing environments like edge and fog computing. The traditional resource scheduling models allocate resources on the basis of the course criteria such as prioritizing applications or tasks. By considering the subsets of data and individual data elements, the fractional selectivity operates at a finer granularity, and results in precise resource allocation. The proposed model concentrates more on data than the tasks or applications, where it considers the data based on the factors like resource requirements, data relevance and importance. This model is vital in fog computing systems where data needs to be processed effectively and generated at the edge.

The suggested model is more efficient in real-time application scenarios with workload variations and changing conditions. Based on the needs of the different sources or data streams, this model dynamically allocates resources. The adaptive nature of fractional selectivity is crucial in managing IoT data and workloads of edge computing. The edge nodes have minimized resources compared to cloud servers in fog computing. The fractional selectivity optimizes the usage of resources by selecting data that should be off-loaded to the cloud and processed locally. The fractional selectivity significantly contributes to latency reduction by selecting the relevant data at the edge. This is necessary in applications like augmented reality, industrial automation, and autonomous vehicles, which need low latency or real-time processing. The fractional selectivity conserves bandwidth by decreasing the data amount which needs to be sent over the network to the cloud. Particularly, it is valuable in application scenarios where the bandwidth of the network is expensive and limited.

Based on the application policies and criteria, the fractional selectivity customizes the decisions of data processing. Each use case and application have its own rules in both data selection and processing that allows for greater adaptability and flexibility. In the context of IoT applications, a massive amount of data is generated from several devices and sensors. The suggested model significantly processes and manages the data that ensures and preserves the most valuable information. The pseudocode of the fractional selectivity model is described in Algorithm 1. The numerical examination of the proposed model is discussed in section 4, and the proposed model's performance is validated in three application scenarios by utilizing six evaluation measures.

Algorithm 1. Pseudocode of the fractional selectivity model

```
Input: workload
Output: offloading workload to virtual machine (result)
Function process workload locally (workload):
    //Process the entire workload locally
    Result=perform local processing (workload)
    Return result
Function offload to fog nodes (workload, fractional selectivity):
    //Determine the portion of the workload to offload based on fractional selectivity
    Offloaded workload=workload×fractional selectivity
    //Offload the workload to fog nodes
    Fog results=offload processing to fog nodes (offloaded workload)
    Return fog results
Function fractional selectivity ():
```

```
//Generate workload
Workload=generate workload ()
//Determine fractional selectivity based on some criteria
Fractional selectivity=determine fractional selectivity ()
//Decide whether to offload to fog or process locally based on fractional selectivity
If fractional selectivity>threshold:
    //Offload a fraction of the workload to fog nodes
    Results=offload to fog nodes (workload, fractional selectivity)
Else:
    //Process the entire workload locally
    Results=process workload locally (workload)
End
```

4. RESULTS AND DISCUSSION

The proposed fractional selectivity model is implemented utilizing Java 1.8 Java Development Kit (JDK), NetBeans integrated development environment (IDE) 8.2, and iFogSim simulator. This model is analyzed on a system featuring Intel i9 processor, 11 GB RTX 2080Ti GPU, 128 GB of RAM, and 1 TB of hard disk. The performance of fractional selectivity is assessed in three application scenarios; virtual reality (VR) game, electroencephalogram (EEG) healthcare, and toy game. The assumed parameters are as follows: processing speed is 4 million instructions per second (MIPS), number of sensors is 7, number of fog devices is 4, and RAM is 1 KB. The proposed model's effectiveness is validated in light of makespan, ARU, LBL, total cost, delay, and energy consumption. The details about the stimulating environment are presented in Table 2.

Table 2. Details about the stimulating environment

Parameters	Value
Time zone	5
Bandwidth	10,000 B/S
Virtual machine model	Xen
Cost	2
Cost per memory	0.1
Cost per storage	0.01
Operating system	Linux
Architecture	X86

4.1. Evaluation measures

The suggested model's efficacy is analyzed utilizing six different evaluation measures which are, total cost, LBL, ARU, makespan, delay, and energy consumption [35]. Makespan represents the time needed to complete all tasks $CT(ti)$, and its mathematical representation is defined in (1)-(3) [36].

$$Makespan = Max(CT(ti)) \tag{1}$$

where:

$$CT(ti) = AT(ti) + TAT(ti) \tag{2}$$

$$TAT(ti) = WT(ti) + BT(ti) \tag{3}$$

Where BT is represented as the burst time, AT is denoted as the arrival time, WT is indicated as the waiting time, and TAT is the turn-around time. Additionally, LBL estimates the load level of fog computing systems. It is computed by dividing the balanced fog servers (BFSs) with the total available fog servers (FSs), as depicted in (4) [37]. On the other hand, ARU is computed by dividing both the BFSs and overloaded fog servers (OFSs) with the total available FSs, which is mathematically stated in (5) [38].

$$LBL = \frac{BFS_S}{FS_S} \times 100\% \tag{4}$$

$$ARU = \frac{(BFS_S + OFSs)}{FS_S} \times 100\% \tag{5}$$

In the context of fog-computing environments, total cost is defined as the comprehensive expenses related with maintaining, operating, and deploying the fog computing infrastructures [39]. The evaluation measure named total cost is mathematically defined in (6).

$$Total\ cost = \sum_{i=0}^{n} CB(ti) + CM(ti) + CP(ti) \quad (6)$$

Where n represents the number of tasks, $CB(ti)$ denotes the cost of bandwidth usage, $CM(ti)$ indicates the cost of memory usage, $CP(ti)$ denotes the processing cost, and ti is the time. Furthermore, the overall power utilized during the execution of tasks is called as energy consumption E, and it is mathematically expressed in (7).

$$E = \sum_{n=1}^{k} e_t(n) + e_e(n) + e_s(n) \quad (7)$$

Where $e_s(n)$ is represented as the energy in sensing for every task, $e_e(n)$ is denoted as the energy in execution, and $e_t(n)$ is indicated as the energy in transmission. In the context of task allocation, delay is represented as the time lag, which occurs during the process of allocating tasks to appropriate resources.

4.2. Quantitative analysis

In this context, several conventional task allocation models, namely GA, PSO, NSGA-II, Bees, and IPM are utilized for assessing the effectiveness of the proposed fractional selectivity model. As shown in Tables 3 and 4, the suggested model has minimal makespan time and delay than the conventional models in all three-application scenarios (VR game, EEG healthcare, and toy game) for varying number of tasks (50, 90, 130, and 150). Particularly, in the EEG healthcare application scenario, the fractional selectivity model achieves the lowest makespan time of 84.62, 146.27, 224.57, and 245.30 milliseconds (ms) for tasks numbering 50, 90, 130, and 150, respectively. Correspondingly, the fractional selectivity model has minimal delay of 84.54, 146.17, 224.48, and 245.20 for tasks numbering 50, 90, 130, and 150 in the EEG healthcare application scenario. In comparison to the conventional task allocation models, the proposed fractional selectivity model is an innovative model for allocating resources in edge and fog computing environments, because it specifically focuses on fine-grained resource allocation, which are valuable to optimize the responsiveness, efficiency, and performance of edge computing applications. The results comparison of various task allocation models in terms of makespan and delay are presented in Figures 3 and 4.

Table 3. Results of various task allocation models by means of makespan

		Makespan (ms)					
Scenarios	Tasks	GA	PSO	Bees	IPM	NSGA-II	Fractional selectivity
EEG healthcare	50	94.10	93.74	89.50	88.61	87.06	84.62
	90	155.03	154.50	149.96	148.83	148.64	146.27
	130	243.18	238.08	228.97	227.75	226.66	224.57
	150	258.23	256.56	251.85	250.92	247.72	245.30
VR game	50	91.25	87.46	86.43	84.81	82.48	80.46
	90	152.27	147.72	146.53	146.16	144.24	142.13
	130	235.96	226.60	225.40	224.19	222.36	220.33
	150	254.46	249.47	248.61	245.36	242.85	240.42
Toy game	50	84.99	84.01	82.74	80.07	78.06	75.90
	90	145.50	144.40	143.84	142.23	139.81	137.36
	130	224.31	222.91	222.04	219.94	218.19	215.95
	150	247.10	246.41	242.97	240.71	238.15	236.02

Table 4. Results of various task allocation models in light of delay

		Delay (ms)					
Scenarios	Tasks	GA	PSO	Bees	IPM	NSGA-II	Fractional selectivity
EEG healthcare	50	94.02	93.67	89.47	88.60	86.97	84.54
	90	154.95	154.47	149.89	148.81	148.61	146.17
	130	243.11	238.07	228.96	227.72	226.62	224.48
	150	258.14	256.49	251.77	250.91	247.70	245.20
VR game	50	91.24	87.41	86.40	84.73	82.42	80.45
	90	152.20	147.70	146.44	146.15	144.14	142.04
	130	235.95	226.52	225.33	224.11	222.35	220.31
	150	254.45	249.39	248.60	245.28	242.79	240.41
Toy game	50	91.16	87.32	86.35	84.70	82.41	80.37
	90	152.19	147.62	146.36	146.05	144.09	142.03
	130	235.88	226.48	225.27	224.03	222.30	220.30
	150	254.38	249.38	248.51	245.23	242.73	240.40

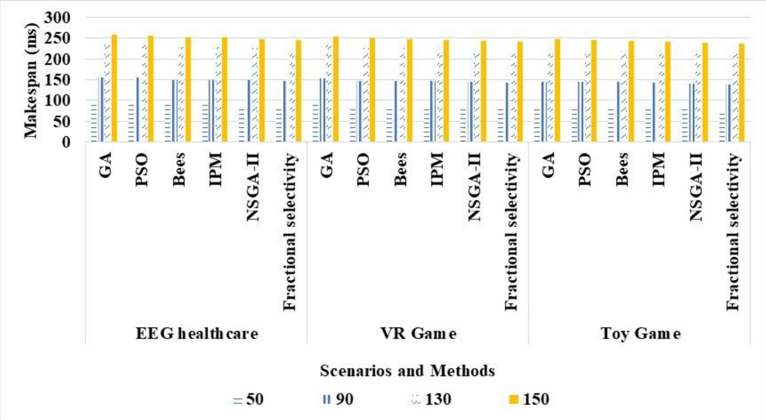

Figure 3. Result comparison of various task allocation models by means of makespan

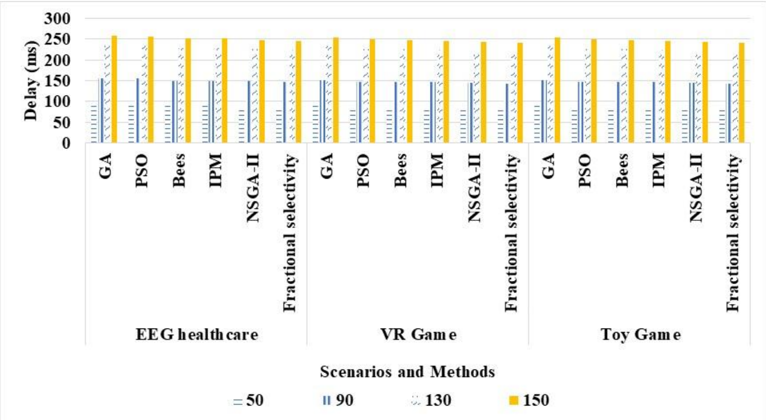

Figure 4. Result comparison of various task allocation models in light of delay

The results of different task allocation models by means of ARU and energy consumption are depicted in Tables 5 and 6. As shown in Tables 5 and 6, the fractional selectivity model has better ARU and energy consumption compared to optimization algorithms GA, PSO, and Bees, but it achieves only a comparable performance when related to the NSGA-II and IPM. Generally, the fractional selectivity model needs an enormous number of resources (processing power and memory) for efficient execution. This limits its applicability in fog and edge computing environments with limited resources. On the other hand, the suggested model is well suited for particular types of workloads and application scenarios. The effectiveness of the proposed model varies based on the nature of the resources and tasks, and it is not applicable for all fog-computing scenarios. The results comparison of six different task allocation models in terms of ARU and energy consumption are depicted in Figures 5 and 6.

Table 5. Results of different task allocation models in light of ARU

		ARU (%)					
Scenarios	Tasks	GA	PSO	Bees	IPM	NSGA-II	Fractional selectivity
EEG healthcare	50	41.68	42.88	50.97	51.91	53.52	51.23
	90	54.71	54.50	53.46	54.66	56.89	54.55
	130	67.19	68.18	69.55	70.55	73.13	70.83
	150	69.52	70.73	81.56	83.17	84.93	82.81
VR game	50	40.63	48.92	49.65	51.41	48.77	46.39
	90	52.14	50.97	52.21	54.87	52.39	50.38
	130	66.08	67.48	68.40	70.73	68.49	66.07
	150	68.23	79.45	81.17	82.54	80.75	78.36
Toy game	50	46.55	47.24	49.07	46.70	44.18	42.01
	90	48.69	49.74	52.49	50.06	48.33	45.91
	130	65.33	66.21	68.24	66.26	63.71	61.23
	150	77.16	78.83	80.41	78.56	75.96	73.70

Table 6. Results of different task allocation models in light of energy consumption

		Energy consumption (Joules)					
Scenarios	Tasks	GA	PSO	Bees	IPM	NSGA-II	Fractional selectivity
EEG healthcare	50	41.60	42.84	50.96	51.83	53.46	51.13
	90	54.70	54.46	53.42	54.60	56.81	54.51
	130	67.11	68.14	69.51	70.51	73.07	70.80
	150	69.47	70.71	81.51	83.12	84.89	82.77
VR game	50	40.56	48.82	49.63	51.38	48.68	46.37
	90	52.09	50.91	52.12	54.83	52.32	50.33
	130	66.01	67.41	68.35	70.65	68.47	66.06
	150	68.23	79.41	81.11	82.50	80.66	78.35
Toy game	50	46.48	47.15	49.01	46.69	44.11	41.92
	90	48.60	49.65	52.41	50.03	48.26	45.82
	130	65.31	66.13	68.24	66.25	63.62	61.14
	150	77.11	78.83	80.38	78.48	75.92	73.67

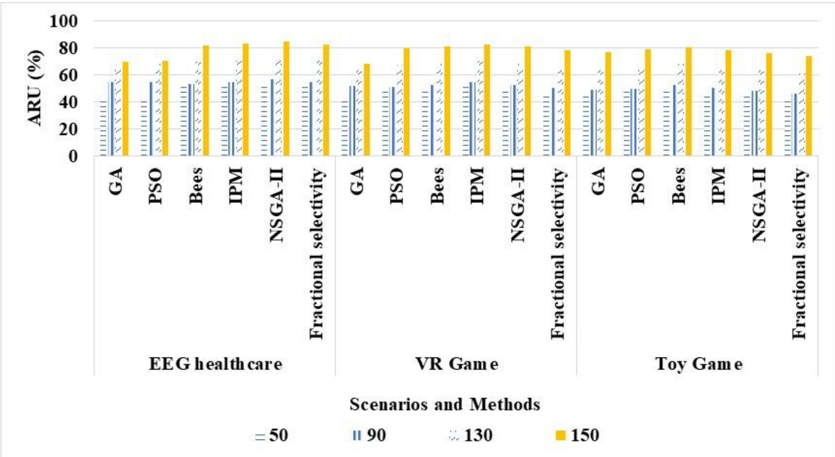

Figure 5. Result comparison of six different task allocation models by means of ARU

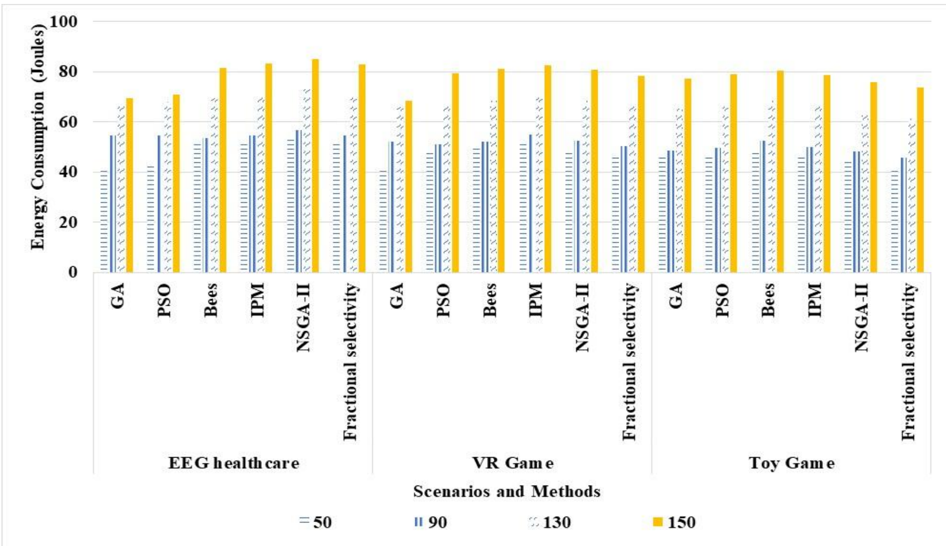

Figure 6. Result comparison of six different task allocation models in light of energy consumption

By inspecting Table 7, similar to ARU, the proposed fractional selectivity model achieves significant LBL compared to GA and PSO. However, it achieves comparable performance with that of the Bees, NSGA-II, and IPM models. The parameters assumed in GA are as follows; crossover function is 0.8, elite count is 2, scaling fraction is rank, stall generations are 50, generations are 230, and population creation is constraint dependence. Additionally, the assumed parameters of PSO are, maximum number of iterations is

100, population size is 50, final inertia weight is 0.2, initial inertia weight is 0.9, maximum particle velocity is 4, and finally c_1 as well as c_2 are 2. Furthermore, the following parameters are assumed in Bees algorithm, initial patch size is 0.1, bees around other selected points are 20, bees around elite points are 50, number of elite sites is 2, maximum number of iterations is 100, and population size is 200. Correspondingly, NSGA-II fixes the following parameters which are, variable type is binary, mutation probability is 34, mutation operator is bit string mutation, crossover probability is one, crossover operator is single point crossover, maximum generation is 200, and population size is 100. The IPM includes respective parameters as; finite difference type is central, lower bound is -15, upper bound is 15, and maximum number of iterations is 100. The results comparison of different task allocation models in terms of LBL is mentioned in Figure 7.

Table 7. Results of various task allocation models by means of LBL

		LBL (%)					
Scenarios	Tasks	GA	PSO	Bees	IPM	NSGA-II	Fractional selectivity
EEG healthcare	50	27.84	29.07	32.66	33.01	33.54	31.22
	90	34.43	34.63	35.10	36.83	37.13	35.01
	130	37.02	42.94	45.91	46.71	47.35	45.08
	150	39.16	44.71	48.14	49.22	50.63	48.22
VR game	50	26.84	30.57	30.98	31.45	28.76	26.76
	90	32.43	32.77	34.79	34.90	32.75	30.56
	130	40.87	43.85	44.59	45.27	42.69	40.25
	150	42.45	45.90	47.22	48.22	45.80	43.54
Toy game	50	28.54	28.77	29.27	26.45	24.46	22.21
	90	30.42	32.74	32.42	30.41	28.27	25.86
	130	41.83	42.40	42.87	40.60	37.92	35.64
	150	43.83	44.87	46.20	43.43	41.07	38.98

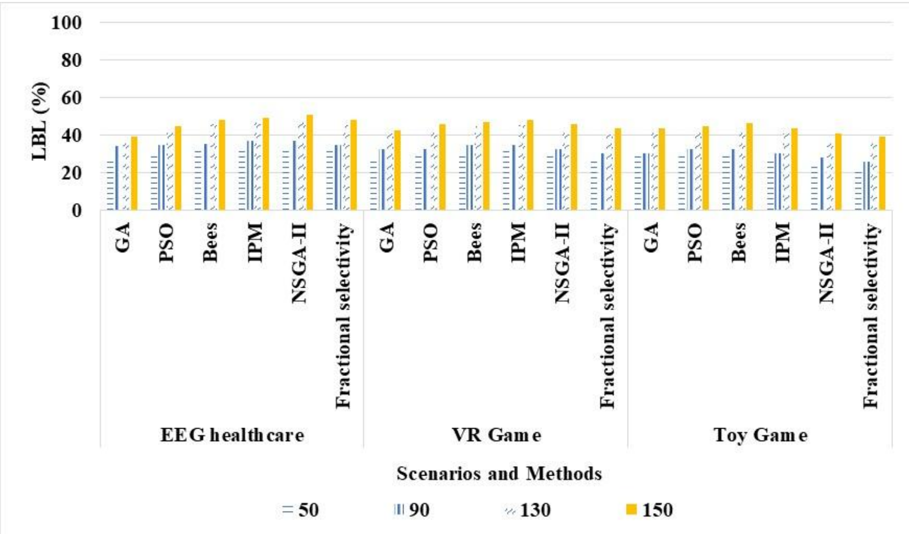

Figure 7. Result comparison of different task allocation models in light of LBL

By viewing Table 8, it is evident that the proposed fractional selectivity model exhibits limited total cost compared to existing task allocation models GA, PSO, NSGA-II, Bees, and IPM. In the context of fog computing, this model aims at reducing total cost and improving resource allocation, as opposed to existing task allocation models. Fractional selectivity model achieves these two objectives by dynamically allocating all computing resources based on the requirements of fog computing applications, and this process results in a cost-effective processing. Unlike traditional task allocation models, the proposed fractional selectivity model assigns resources on-demand, and hence reduces costs and minimizes wastages in fog computing systems. Additionally, it performs scalable resource allocation, which ensures that fog nodes significantly handle increasing workloads without accumulating high costs. The results comparison of six different task allocation models by means of total cost is graphically represented in Figure 8.

Table 8. Results of different task allocation models in terms of total cost

Scenarios	Tasks	GA	PSO	Bees	IPM	NSGA-II	Fractional selectivity
		Total cost					
EEG healthcare	50	3070.58	2994.76	2793.61	1369.50	1360.84	1358.61
	90	3068.66	2991.71	2790.96	1372.70	1362.62	1360.37
	130	3065.62	2987.57	2786.51	1372.76	1364.78	1362.55
	150	3061.93	2984.91	2782.92	1375.71	1367.77	1365.71
VR game	50	2992.57	2791.55	1367.10	1358.36	1356.58	1354.58
	90	2989.56	2788.65	1370.68	1360.17	1358.16	1355.89
	130	2985.25	2784.14	1370.59	1362.72	1360.54	1358.09
	150	2982.52	2780.58	1373.32	1365.59	1363.70	1361.61
Toy game	50	2789.35	1364.90	1356.23	1354.33	1352.27	1350.02
	90	2786.60	1368.25	1357.78	1355.98	1353.65	1351.33
	130	2782.13	1368.19	1360.46	1358.24	1355.76	1353.27
	150	2778.16	1371.20	1363.35	1361.42	1359.46	1357.36

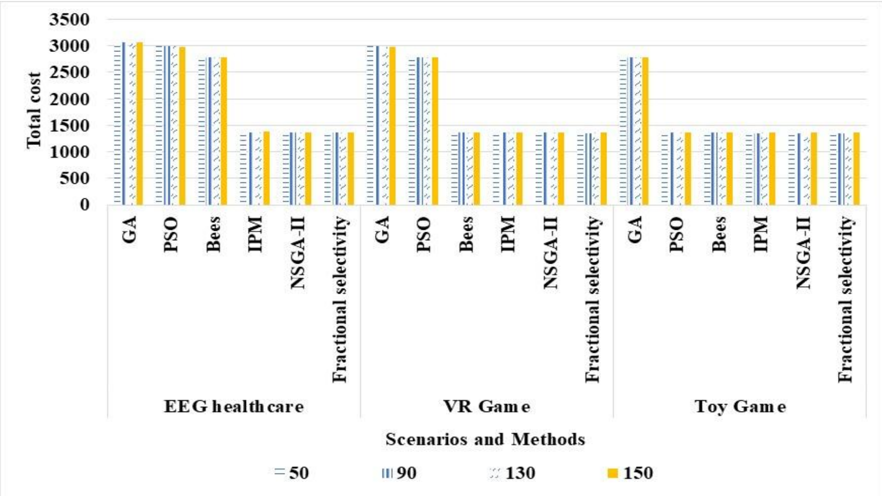

Figure 8. Result comparison of different task allocation models by means of total cost

4.3. Discussion

As depicted in Tables 3 to 8, the proposed fractional selectivity model offers more benefits in fog computing systems than the traditional task allocation models. Generally, the fractional selectivity model splits and executes the tasks in multiple fog nodes that results in better resource utilization with reduced resource wastage. The system becomes more fault-tolerant by distributing tasks across several fog nodes. Based on the fog nodes' current workload and their capability, the fractional selectivity model dynamically distributes tasks among fog nodes. This process results in better LB and prevents fog nodes from being overloaded. Additionally, the fractional selectivity model helps in minimizing energy consumption and latency in fog nodes, by efficiently distributing tasks, and is more scalable when the workload increases. Particularly in task allocation, the fractional selectivity model offers higher flexibility because it has better adaptation to changing requirements and workloads. The suggested model improves QoS in fog computing systems by ensuring that all tasks are assigned to fog nodes with proper resources. In conclusion, the proposed fractional selectivity based-task allocation model superiorly improves the reliability, flexibility, and efficiency of fog computing systems, and is cost-efficient as it minimizes the operational costs, the use of additional hardware resources required for maintenance, and the energy consumption.

5. CONCLUSION

In this article, a novel fractional selectivity model is proposed in fog computing environments for efficient task allocation. In a fog computing system, the fractional selectivity model initially splits a single task into different portions or fractions. Furthermore, these fractions are allocated to multiple resources or fog nodes for better execution. Related to the conventional binary task allocation models, the proposed fractional selectivity model provides higher optimization possibilities and flexibility for task allocation, especially in fog computing systems. In this article, the proposed fractional selectivity model's efficiency is analyzed using six dissimilar evaluation measures, namely delay, energy consumption, total cost, LBL, ARU, and makespan. In comparison to the traditional task allocation models such as GA, PSO, NSGA-II, Bees, and IPM, the fractional selectivity model is superior in reducing total cost and makespan, and improving LBL and ARU percentages, in three different application scenarios. However, the proposed fractional selectivity model requires a vast amount of resources for better execution, and it is suited only for specific types of application scenarios and workloads. Therefore, as a future extension, an effective population-based optimization algorithm will be integrated with the proposed fractional selectivity model, to further enhance the performance of task allocation in all types of application scenarios and workloads.

ACKNOWLEDGEMENTS

Authors would like to thank M S Ramaiah Institute of Technology, Bengaluru for providing the infrastructure required for the research and in particular the access for the journal articles.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

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

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Prasanna Kumar Kannughatta Ranganna	✓	✓	✓	✓	✓	✓		✓	✓	✓				✓
Siddesh Gaddadevara Matt		✓	✓	✓			✓			✓		✓		
Ananda Babu Jayachandra		✓		✓		✓		✓	✓	✓	✓	✓	✓	✓
Vasantha Kumara Mahadevachar	✓		✓	✓			✓			✓	✓		✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The authors confirm that the data supporting the findings of this study are available within the article.

REFERENCES

[1] M. Ramya Devi and S. Lokesh, "Intelligent accident detection system by emergency response and disaster management using vehicular fog computing," *Automatika*, vol. 65, no. 1, pp. 117–129, Jan. 2024, doi: 10.1080/00051144.2023.2288483.

[2] A. Elhadad, F. Alanazi, A. I. Taloba, and A. Abozeid, "Fog computing service in the healthcare monitoring system for managing the real-time notification," *Journal of Healthcare Engineering*, vol. 2022, pp. 1–11, Mar. 2022, doi: 10.1155/2022/5337733.

[3] U. Tadakamalla and D. A. Menasce, "Autonomic resource management for fog computing," *IEEE Transactions on Cloud Computing*, vol. 10, no. 4, pp. 2334–2350, Oct. 2022, doi: 10.1109/TCC.2021.3064629.

[4] C. Yi, J. Cai, K. Zhu, and R. Wang, "A queueing game based management framework for fog computing with strategic computing speed control," *IEEE Transactions on Mobile Computing*, vol. 21, no. 5, pp. 1537–1551, May 2022, doi: 10.1109/TMC.2020.3026194.

[5] K. R. Kiran, D. L. S. Kumar, V. E. Jyothi, N. H. H. Cuon, and N. H. Ha, "An advanced ensemble load balancing approach for fog computing applications," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 2, pp. 1825-1833, Apr. 2024, doi: 10.11591/ijece.v14i2.pp1825-1833.

[6] S. S. Nejad, A. Khademzadeh, A. M. Rahmani, and A. Broumandnia, "Resource allocation for fog computing based on software-defined networks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 6, pp. 7099-7107, Dec. 2023, doi: 10.11591/ijece.v13i6.pp7099-7107.

[7] H. D. K. Al-Janabi *et al.*, "D-BlockAuth: an authentication scheme-based dual blockchain for 5G-assisted vehicular fog computing," *IEEE Access*, vol. 12, pp. 99321–99332, 2024, doi: 10.1109/ACCESS.2024.3428830.

[8] Z. Shamsa, A. Rezaee, S. Adabi, and A. M. Rahmani, "A decentralized prediction-based workflow load balancing architecture for cloud/fog/IoT environments," *Computing*, vol. 106, no. 1, pp. 201–239, Jan. 2024, doi: 10.1007/s00607-023-01216-3.

[9] Z. G. Al-Mekhlafi *et al.*, "Lattice-based cryptography and fog computing based efficient anonymous authentication scheme for 5G-assisted vehicular communications," *IEEE Access*, vol. 12, pp. 71232–71247, May 2024, doi: 10.1109/ACCESS.2024.3402336.

[10] F. R. Shahidani, A. Ghasemi, A. T. Haghighat, and A. Keshavarzi, "Task scheduling in edge-fog-cloud architecture: a multi-objective load balancing approach using reinforcement learning algorithm," *Computing*, vol. 105, no. 6, pp. 1337–1359, Jun. 2023, doi: 10.1007/s00607-022-01147-5.

[11] C. Liu, J. Wang, L. Zhou, and A. Rezaeipanah, "Solving the multi-objective problem of IoT service placement in fog computing using cuckoo search algorithm," *Neural Processing Letters*, vol. 54, no. 3, pp. 1823–1854, Jun. 2022, doi: 10.1007/s11063-021-10708-2.

[12] B. Wu, X. Lv, W. D. Shamsi, and E. G. Dizicheh, "Optimal deploying IoT services on the fog computing: a metaheuristic-based multi-objective approach," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 10010–10027, Nov. 2022, doi: 10.1016/j.jksuci.2022.10.002.

[13] Z. G. Al-Mekhlafi *et al.*, "Oblivious transfer-based authentication and privacy-preserving protocol for 5G-enabled vehicular fog computing," *IEEE Access*, vol. 12, pp. 100152–100166, 2024, doi: 10.1109/ACCESS.2024.3429179.

[14] M. Ghobaei-Arani and A. Shahidinejad, "A cost-efficient IoT service placement approach using whale optimization algorithm in fog computing environment," *Expert Systems with Applications*, vol. 200, Aug. 2022, doi: 10.1016/j.eswa.2022.117012.

[15] 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, May 2022, doi: 10.1016/j.jnca.2022.103333.

[16] M. Kaur and R. Aron, "An energy-efficient load balancing approach for scientific workflows in fog computing," *Wireless Personal Communications*, vol. 125, no. 4, pp. 3549–3573, Aug. 2022, doi: 10.1007/s11277-022-09724-9.

[17] S. Gupta and N. Singh, "Fog-GMFA-DRL: enhanced deep reinforcement learning with hybrid grey wolf and modified moth flame optimization to enhance the load balancing in the fog-IoT environment," *Advances in Engineering Software*, vol. 174, Dec. 2022, doi: 10.1016/j.advengsoft.2022.103295.

[18] F. M. Talaat, H. A. Ali, M. S. Saraya, and A. I. Saleh, "Effective scheduling algorithm for load balancing in fog environment using CNN and MPSO," *Knowledge and Information Systems*, vol. 64, no. 3, pp. 773–797, Mar. 2022, doi: 10.1007/s10115-021-01649-2.

[19] F. M. Talaat, M. S. Saraya, A. I. Saleh, H. A. Ali, and S. H. Ali, "A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environment," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 4951–4966, Nov. 2020, doi: 10.1007/s12652-020-01768-8.

[20] M. Kaur and R. Aron, "FOCALB: fog computing architecture of load balancing for scientific workflow applications," *Journal of Grid Computing*, vol. 19, no. 4, Dec. 2021, doi: 10.1007/s10723-021-09584-w.

[21] M. K. Hussein and M. H. Mousa, "Efficient task offloading for IoT-based applications in fog computing using ant colony optimization," *IEEE Access*, vol. 8, pp. 37191–37201, 2020, doi: 10.1109/ACCESS.2020.2975741.

[22] J. Singh, P. Singh, E. M. Amhoud, and M. Hedabou, "Energy-efficient and secure load balancing technique for SDN-enabled fog computing," *Sustainability*, vol. 14, no. 19, Oct. 2022, doi: 10.3390/su141912951.

[23] I. Z. Yakubu and M. Murali, "An efficient meta-heuristic resource allocation with load balancing in IoT-fog-cloud computing environment," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 3, pp. 2981–2992, Mar. 2023, doi: 10.1007/s12652-023-04544-6.

[24] D. Baburao, T. Pavankumar, and C. S. R. Prabhu, "Load balancing in the fog nodes using particle swarm optimization-based enhanced dynamic resource allocation method," *Applied Nanoscience*, vol. 13, no. 2, pp. 1045–1054, Feb. 2023, doi: 10.1007/s13204-021-01970-w.

[25] D. Javaheri, S. Gorgin, J.-A. Lee, and M. Masdari, "An improved discrete Harris hawk optimization algorithm for efficient workflow scheduling in multi-fog computing," *Sustainable Computing: Informatics and Systems*, vol. 36, Dec. 2022, doi: 10.1016/j.suscom.2022.100787.

[26] A. Kishor and C. Chakarbarty, "Task offloading in fog computing for using smart ant colony optimization," *Wireless Personal Communications*, vol. 127, no. 2, pp. 1683–1704, Nov. 2022, doi: 10.1007/s11277-021-08714-7.

[27] S. P. Singh, "Effective load balancing strategy using fuzzy golden eagle optimization in fog computing environment," *Sustainable Computing: Informatics and Systems*, vol. 35, Sep. 2022, doi: 10.1016/j.suscom.2022.100766.

[28] B. V. Natesha and R. M. R. Guddeti, "Adopting elitism-based genetic algorithm for minimizing multi-objective problems of IoT service placement in fog computing environment," *Journal of Network and Computer Applications*, vol. 178, Mar. 2021, doi: 10.1016/j.jnca.2020.102972.

[29] M. Bey, P. Kuila, B. B. Naik, and S. Ghosh, "Quantum-inspired particle swarm optimization for efficient IoT service placement in edge computing systems," *Expert Systems with Applications*, vol. 236, Feb. 2024, doi: 10.1016/j.eswa.2023.121270.

[30] K. Gai, X. Qin, and L. Zhu, "An energy-aware high performance task allocation strategy in heterogeneous fog computing environments," *IEEE Transactions on Computers*, vol. 70, no. 4, pp. 626–639, Apr. 2021, doi: 10.1109/TC.2020.2993561.

[31] M. Iyapparaja, N. K. Alshammari, M. S. Kumar, S. S. R. Krishnan, and C. L. Chowdhary, "Efficient resource allocation in fog computing using QTCS model," *Computers, Materials & Continua*, vol. 70, no. 2, pp. 2225–2239, 2022, doi: 10.32604/cmc.2022.015707.

[32] J. Gu, J. Mo, B. Li, Y. Zhang, and W. Wang, "A multi-objective fog computing task scheduling strategy based on ant colony algorithm," in *2021 IEEE 4th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, Sep. 2021, pp. 12–16, doi: 10.1109/ICISCAE52414.2021.9590674.

[33] S. Vemireddy and R. R. Rout, "Fuzzy reinforcement learning for energy efficient task offloading in vehicular fog computing," *Computer Networks*, vol. 199, Nov. 2021, doi: 10.1016/j.comnet.2021.108463.

[34] X. Hou, Z. Ren, J. Wang, S. Zheng, W. Cheng, and H. Zhang, "Distributed fog computing for latency and reliability guaranteed swarm of drones," *IEEE Access*, vol. 8, pp. 7117–7130, 2020, doi: 10.1109/ACCESS.2020.2964073.

[35] V. Gowri and B. Baranidharan, "An energy efficient and secure model using chaotic levy flight deep Q-learning in healthcare system," *Sustainable Computing: Informatics and Systems*, vol. 39, Sep. 2023, doi: 10.1016/j.suscom.2023.100894.

[36] F. M. Talaat, "Effective deep Q-networks (EDQN) strategy for resource allocation based on optimized reinforcement learning algorithm," *Multimedia Tools and Applications*, vol. 81, no. 28, pp. 39945–39961, Nov. 2022, doi: 10.1007/s11042-022-13000-0.

[37] F. M. Talaat, "Effective prediction and resource allocation method (EPRAM) in fog computing environment for smart healthcare system," *Multimedia Tools and Applications*, vol. 81, no. 6, pp. 8235–8258, Mar. 2022, doi: 10.1007/s11042-022-12223-5.
[38] S. S. Tripathy, D. S. Roy, and R. K. Barik, "M2FBalancer: A mist-assisted fog computing-based load balancing strategy for smart cities," *Journal of Ambient Intelligence and Smart Environments*, vol. 13, no. 3, pp. 219–233, 2021, doi: 10.3233/AIS-210598.
[39] S. Alshathri, F. M. Talaat, and A. A. Nasr, "A new reliable system for managing virtual cloud network," *Computers, Materials & Continua*, vol. 73, no. 3, pp. 5863–5885, 2022, doi: 10.32604/cmc.2022.026547.

BIOGRAPHIES OF AUTHORS

Prasanna Kumar Kannughatta Ranganna is currently pursuing Ph.D. at Ramaiah Institute of Technology, Bangalore affiliated to Visvesvaraya Technological University Belagavi. He obtained his masters of technology in Software Engineering from Sri Jayachamarajendra College of Engineering, Mysore and is working at Siddaganga Institute of Technology, Tumkur since February 2006. His research focus is towards fog computing, edge computing, IoT, and cloud computing. He can be contacted at email: prasanna.kghatta@sit.ac.in.

Siddesh Gaddadevara Matt is working as Professor in the Department of CSE (AI & ML) at M S Ramaiah Institute of Technology, Bangalore. His research interest is focused on IoT, cloud computing, fog computing, and machine learning. He can be contacted at email: siddeshgm@msrit.edu

Dr. Ananda Babu Jayachandra is working as Associate Professor in the Department of Information Science and Engineering at Malnad College of Engineering Hassan. He has guided 6 research scholars for their doctoral degree. His research interests span across IoT, machine learning, image processing, and computer vision. He has published more than 25 scholarly articles in reputed journals and has received more than 100 citations so far. He can be contacted at email: abj@mcehassan.ac.in.

Dr. Vasantha Kumara Mahadevachar holds a Doctoral Degree in Computer Science and Engineering from VTU, Belagavi. He is working as Assistant Professor in computer science and engineering at Government College of Engineering Hassan. His research interests span cross computer vision, industrial IoT and machine learning applications. He has published more than 10 publications journals/conferences of high quality. He can be contacted at email: cmn.vasanth@gmail.com.