International Journal on Advanced Science Engineering Information Technology

Vol.12 (2022) No. 6
ISSN: 2088-5334

A Fog Computing Framework in IoT Healthcare Environment: Towards A New Method Based on Tasks Significance

Abeer M. Shanshool [a,*], Noor Alhuda F. Abbas [a]

[a] *Department of Computer Technologies Engineering, AL-Esraa University Collage, Baghdad, Iraq*
*Corresponding author: *abeer@esraa.edu.iq*

***Abstract*—The Internet-of-Things (IoT) is an important technology and is considered the future of the Internet. Healthcare is described as one of the important areas in IoT used for remote patient monitoring. Real-time remote monitoring health applications are important as delays in data transfer between the cloud, and the application may be unacceptable. Fog computing refers to a geographically distributed computing system with several devices connected to the same network to achieve flexible and collaborative computation, storage, and communication services. Fog computing is mainly used for efficient data processing between sensors and cloud computing as it reduces the volume of data exchanged between sensors and the cloud, thereby improving the whole system's efficiency. Wireless sensor networks (WSN) are also used in health monitoring systems to simultaneously transfer huge data volumes (of different priority levels and length values) to the fog computing system. Hence, there is a need to appropriately implement a task scheduling mechanism that can accurately prioritize tasks irrespective of their length. This study aims to systematically review the existing fog computing technologies in the Internet of things HealthCare (IoTH) systems and improve the performance of the available static task scheduling algorithms using the Tasks Classification (TC) method where task importance is paramount. The performance of the suggested approach was evaluated based on the Max-Min scheduling algorithm (SA).**

***Keywords*— Fog computing; edge computing; cloud computing; AI; IoT healthcare.**

Manuscript received 14 Aug. 2021; revised 17 Apr. 2022; accepted 6 Jun. 2022. Date of publication 31 Dec. 2022.

I. INTRODUCTION

Presently, cloud computing is a basic aspect of most human life activities even though it still faces certain challenges, such as high delays, which are bottlenecks to most IoT services that demand real-time responses [1]. It is also affecting its adoption in industrial control systems where a low delay is only required in response times. The Fog computing was suggested for application at three networking levels: (i) data collection from the edge devices; (ii) connection of numerous devices to a network and for data transfer between the devices; (iii) processing of the collected data from the devices within a second and making the necessary decisions [2]. Fog computing is used for edge devices' computation, communication, and storage capabilities. Thereby facilitating security, privacy, mobility, and low latency [2]. Fog computing was designed to meet applications requiring real-time (RT) response and low complexity, like the IoT applications in healthcare [3]. The huge volume of data generated by healthcare applications can be efficiently stored in the cloud instead of storing them on computing devices and other storage devices that have limited storage spaces [4]. The large volume of data generated during healthcare diagnosis requires efficient storage and retrieval. Sometimes, the streaming of such data may be considered in e-health applications in times of real-time demand [5]. Considering these requirements, fog computing has commonly chosen in designing Healthcare apps as these apps are sensitive to latency and generate large volumes of data [6]. Fog computing can be employed in monitoring early people even when they are at their private homes using the home nursing system [7], [8].

Fog computing infrastructure can be built from one fog node or numerous computing nodes by connecting them systematically [9]. Connecting these fog computing nodes improves scalability, elasticity, redundancy, and computing efficiency. In fog computing, many fog nodes can be added to the system when more computing is needed. These attributes are among the basic requirement of most healthcare apps [10], [11]. One can rely on the good attributes of fog computing to provide support for numerous healthcare applications. This paper presents a fog computing-based approach designed to improve the features (performance, reliability, interoperability) of the IoT architectures in the healthcare system. This study made the following contributions:

- Highlighting the limitations facing IoT systems in the healthcare system.
- Related work of the fog computing in IoTH systems.
- Development of a mobility support system using fog computing for seamless connection of mobile sensors with cloud computing to ensure efficient data collection and processing and reduce the volume of data exchanged between sensors and the cloud.

A. Fog Computing in IOTH System

The fog computing based on IoT Healthcare (IoTH) has three major layers, as seen in figure 1. These layers are:

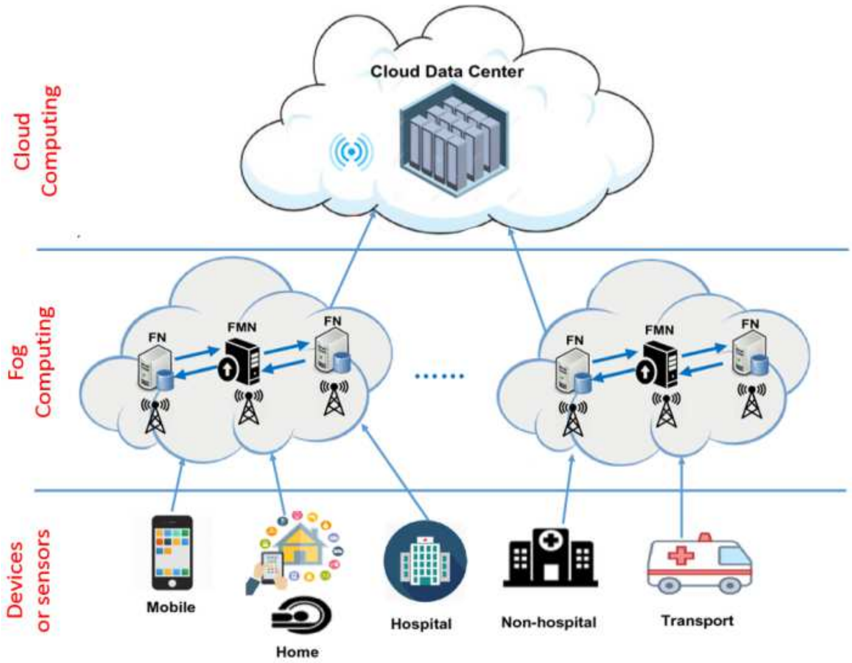

Fig. 1 Framework of IoT/fog computing-based healthcare systems

1) Devices or Sensors: This layer collects healthcare data from the devices and sensors for onward transmission to the FC during Wi-Fi or cellular networks for task processing and execution. These devices and sensors are distributed to the patient's body to monitor their health status [8].

2) Fog Computing (FC): The main benefit of this layer is a connection between the first layer and the third (Sensors and Cloud Computing). This layer receives the data from the sensors/devices and then analyzes and processes it. The processed and analyzed data is transmitted in real-time to notify the user about the patient's health status [9]. On another side, this layer is linked to the last layer (Cloud computing) to analyze, store, and compile the patients' health records [10].

3) Cloud Computing Layer: This layer stores, processes, and executes tasks that cannot be performed by the second layer (FC) [12], [13], [14]. Information details and the patient's status are sent to this layer for needed actions. The fog layer can retrieve the information stored in the cloudayer to be studied and analyzed [15].

B. Related Works

Edge Mesh has been presented as one of the new computing paradigms for distributing decision-making tasks amongst fog nodes and smart gateways [16]. In this system, personal gateways on the patient's side serve as the intermediate node (called the fog node) for processing the patient's health data. An algorithm has been developed for resource sharing between fog nodes [17]. Some studies have also reported the use of smart gateways in fog computing, where they serve as the fog node in healthcare apps. A model has been advanced with 2 algorithms; the 1st algorithm is for picking a fog when a user is at an overlapping part of the fog, and whether one is for solving situations involving location changes by the user [18]. Dynamic resource allocation using fog computing with smart gateways (Micro Data-center) has been suggested [19].

Earlier, the use of personal gateways as intermediate nodes of fog which are utilized between the healthcare cloud and IoT devices, has been reported [20]. An algorithm for clustering all cells could improve resource sharing between them [21]. In comparison, another optimal resource-sharing approach could maximize the corresponding utility [22]. A hybrid system that combines Mobile Edge Computing (MEC) and Software-Defined Systems (SDSys) for constructing ubiquitous MEC where many local controllers are connected by a global controller [23]. Cloud computing is used to mitigate resource management via sharing device resources among users [24].

The edge nodes may be sufficiently managed with an IoT-cloud model called Stack 4 Things as it allows allocating resources closer to reduce processing time and latency [25]. The medical devices were based on a security provisioning model (AZSPM) in fog platforms [26]. Cloud at edge or fog, a virtual platform, was considered for serving on-demand execution platforms of micro-services near devices and differed from micro-services execution in the actual device [27]. This system permits the placement of the API gateway within the micro-services to affect connectivity between the gateway distribution and aggregation function. A fog computing-based protocol of triparty" with one-round key authentication [28]. This protocol relies on bilinear pairing cryptography for the generation of the session that ensures secure communication between the communicating parties. A new framework that facilitates remote sensing and monitoring in real-time for making prognosis and diagnosis was presented [29]. The major factors in managing IoT systems in health care are computation offloading, interoperability, and load balancing/distribution.

A comparison of the recently developed frameworks (presented in the previous table) showed that no framework had addressed the issue of computation offloading and the "constraints imposed on all factors of resource management. Hence, there is a need for a model that can address the fundamental factors of resource management in Healthcare IoT systems. This section focuses on studies that proposed and implemented fog computing-based systems in healthcare applications.

A better service provisioning based on the fog platform supports smart living data flow analysis using Foglet [30]. This system builds a subnetwork by connecting smart objects to a Fog Edge Node (FEN). A fog computing system has been developed for the data monitoring of patient health for ambient assistance [31]. This system has a group of nodes connected to the cloudlet to reduce the burden on the communication infrastructure. A simulation process was conducted using a discrete event system specification (DEVS) that provides lower waiting times by connecting fogs to a broker [32].

An end-to-end security scheme was presented that implemented a network of inter-linked smart gateways [33]. The proposed system reduced communication traffic to 26 % while the communication latency between end-users and smart gateways was reduced to 16 %. Veterinary healthcare with fog computing was presented [34]. It is comprised of several nodes for voluntary storage and computation work. The performance of the healthcare monitoring system can be improved by exploiting fog computing at smart gateways to provide advanced services at the end of the network [35]. The IoTH concept has been demonstrated for sensor data computing [36]. This work used different techniques such as blockchain, fog computing, and IoT. This technique facilitates intelligent computation at small autonomous units, such as edge clouds or smartphones.

A "fog computing-based medical warning system where decision-making is achieved via passing the hypothesis function described in the Analyze component to the Plan component in the gateway [37]. This reinforces the system for the facilitation of efficient decision-making on local networks while data collection is done in a wireless manner using heterogeneous sensor network devices before being forwarded to the cloud server via the gateway. A fog computing interface (FIT) low-power system was proposed as a smart gateway for clinical speech data processing [21]. Before the transfer of the speech features to secure cloud storage, it is first gathered, stored, and processed by the FIT. The proposal for an inexpensive healthcare system has been made for remote monitoring, analyses, and notifications of the ECG [38]. This system is comprised of energy-aware sensor nodes and a fog layer that relies on IoT.

Some studies have also presented fog computing-based healthcare applications that did not rely on the use of smart gateway and shared nodes techniques. For instance, smartphone-based systems called Emergency Help Alert Mobile Cloud (EHAMC) is a way of communication between different units during an emergency; this system is based on fog services for the performance of the offloading and preprocessing tasks [39]. A new Internet of health things (IoHT) method was developed to solve healthcare limitations [40]. Here, the patient's quality of life depends on the extent of the patient's mobility. A Fog-Healthcare architecture and blockchain technology were presented for tracking household activities with more security [41]. The system contained several monitoring devices and edges for cloud computing and storage. The edge clouds with containers rely on edge cloud architecture PaaS and service orchestration to manage and coordinate applications rather than using virtual machines [42]. A previous study has introduced fog computing and combined it with Deep Learning models [43].

II. Materials and Methods

A. Proposed Scheduling Method

This work proposes a TC-based approach to IoTH task scheduling in FC by considering the task's importance. This proposal aims to improve the performance of FC and overcome some of the problems of the current task scheduling systems. The VMs were partitioned into a denomination, and a classification method was applied for the tasks in consideration of the importance of the tasks. The classification can be defined as the "systematic arrangements in the groups or categories based on the criteria that have been established." Task classification in TC uses the significance of the tasks, which is considered to classify them into groups before assigning them to the most suitable VMs classes. The creation of the VMs classes is based on categorizing the VM list into numerous denominations, where "VM classed set = tasks classification (TC) set."

IoTH task may be divided into three categories or classes. The first is critical tasks, which are highly sensitive to any delay and are needed to be transmitted immediately in real-time upon alerts or notifications. The second is important tasks, which are the ones that need correct and timely information access. The last is general tasks encompasses tasks that any delay in information access may not cause any problems.

B. IoTH Tasks Scheduling with TC Using Tasks Significant.

This method considers the importance of tasks during the scheduling of IoT tasks rather than the length of tasks. There are two major stages of the proposed method as mentioned in Table 1.

TABLE I
STAGES OF THE PROPOSED METHOD

Stages	Description	Categories
Stage 1	Consider three classes of tasks that are partitioned in consideration of their importance; three classes are created to cluster the high, medium, and low importance tasks into specific classes. The three categories are described.	• This class contains the tasks that are of low importance. The (Max, Min) scheduling algorithm is used to sort the tasks according to their importance. • This class receives tasks that are of medium importance. The (Max, Min) scheduling algorithm is also used to sort the tasks based on their importance. • This class receives tasks of high importance. The (Max, Min) scheduling algorithm is used to sort the tasks based on their importance.
Stage 2	The VM List is also partitioned to three categories (VMs categories set = tasks classification (TC) set) and any category specifically serves a specific task type based on importance.	• contains a set of low capability and low-performance VMs and as such, these VMs only receive & serve tasks of low priority. Tasks assignment to VMs is based on the VM with the lowest workload. • contains a set of medium capability and medium performance VMs which specifically receive & serve tasks of medium priority. Tasks assignment to VMs is based upon VMs with the lowest work-load. • contains a group of the high capability and high achievement VM that specifically receive & serve tasks of high priority. Tasks assignment to VMs is using the VM with the lowest workload.

The TC Using tasks significant is performed thus: The arriving tasks are classified into 3 classes in consideration of their importance. These classes are sorted using the (Max, Min). Each task class is allocated to the appropriate VMs category for execution based on the importance.

C. IoTH tasks scheduling Using Tasks Significant

Figure 2 is the flowchart of the proposed approach to task scheduling. it explains the process of TCVC-based scheduling of IoTH tasks based on tasks significant and using the (Max, Min) in FC platforms.

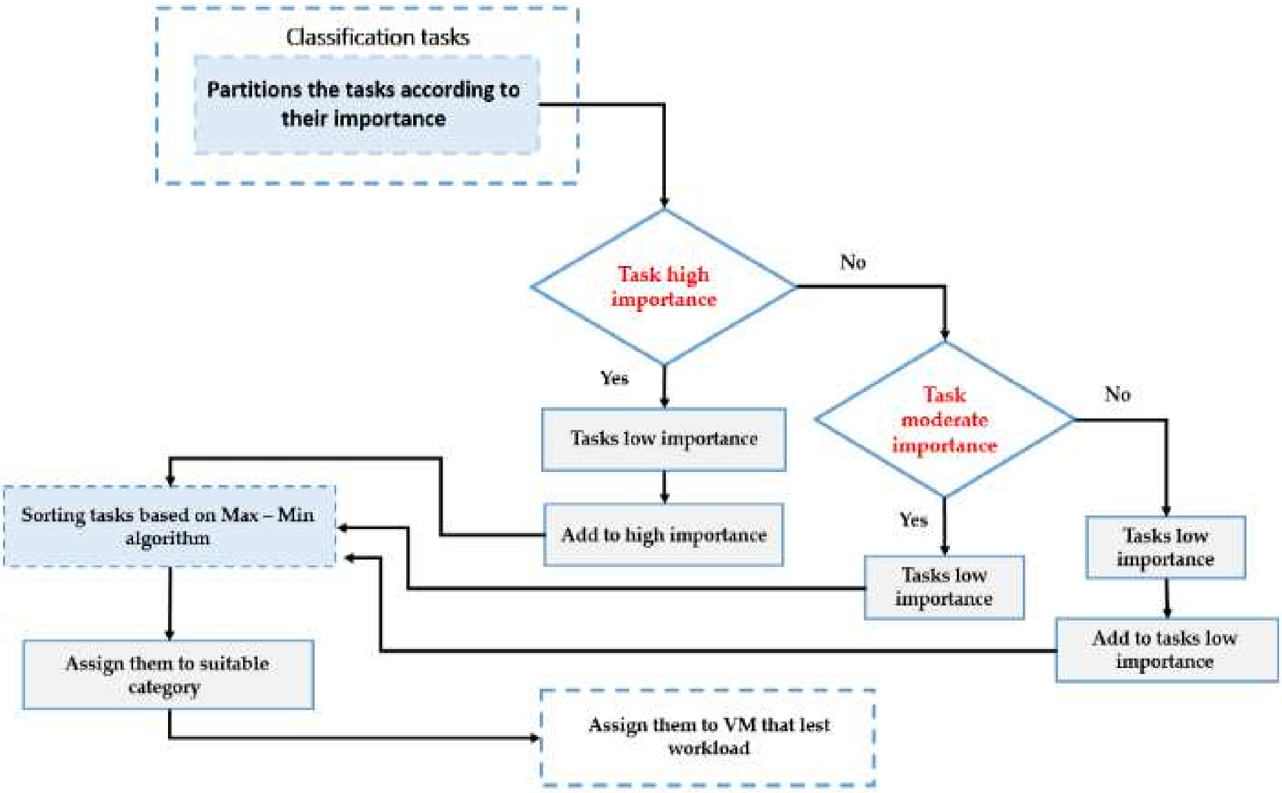

Fig. 2 Flowchart of TC based upon task significance

D. Impacts of TC Using the Significant of (Max, Min) Method of Scheduling

The efficiency of suggested work was evaluated on "Max-Min and after that it has been compared for performance with and with no developed approach. The efficiency was evaluated based on certain metrics, such as ET, FT and WT of significant tasks. For each task, the relative significance was determined using the suggested basic scale for showing the quantitative numbers from Table 2 ".

TABLE II

BASIC MEASUREMENTS FOR SCHEDULING OF TASKS USING TASK SIGNIFICANT

Length Task	Primacy	Tasks No.
LT = 1000	Low task importance	LT11, LT13, LT5, LT1
LT = 20000	Low task importance	LT11, LT13, LT5, LT1
LT = 2000	Low task importance	LT11, LT13, LT5, LT1
LT = 4000	Moderate task importance	LT12, LT6, LT7, LT4
LT = 10000	Moderate task importance	LT12, LT6, LT7, LT4
LT = 90000	Moderate task importance	LT12, LT6, LT7, LT4
LT = 14000	High task importance	LT9, LT14, LT2, LT15, LT3, LT8
LT = 24000	High task importance	LT9, LT14, LT2, LT15, LT3, LT8
LT= 100000	High task importance	LT9, LT14, LT2, LT15, LT3, LT8
LT = 4000	High task importance	LT9, LT14, LT2, LT15, LT3, LT8
LT = 80000	High task importance	LT9, LT14, LT2, LT15, LT3, LT8

E. Max-Min Work Mechanism

Tasks are sorted from the largest to the smallest using the completion time when using MAX-MIN alone before being assigned to the VMs with the least overall ET in the VMs list. As seen in Table 3, small tasks are kept waiting in list for larger tasks to be completed irrespective of the priority of the small ones.

TABLE III

MAX-MIN METHOD OF TASK SCHEDULING

Tasks	Lengths	VMs	WT	ET	FT
t1	1000000	VM 6	0	40	40
t8	100000	VM 5	0	40	40
t7	90000	VM 4	0	60	60
t15	80000	VM 3	0	73.33	73.33
t2	70000	VM 2	0	140	140
t14	25000	VM 1	0	50	50
t13	20000	VM 6	40	8	48
t9	15000	VM 5	40	6	46
t6	10000	VM 4	60	6.67	66.67
t3	5000	VM 3	73.33	3.33	76.66
t12	4000	VM 2	140	8	148
t5	3000	VM 1	50	6	56
t11	2000	VM 6	48	0.8	48.8
t4	1000	VM 5	46	0.4	46.4
t10	1000	VM 4	66.67	0.4	67.07

F. MAX-MIN with TC using Tasks Significant Work

The (Max-Min) algorithm with suggested work sorts the tasks into three portions as follows:

- The 1st part denotes category 1 and gives high-importance tasks to VM-6 and VM-5."
- "The 2nd part is category 2 and gives medium importance tasks to the VM-4 and VM-3."
- The 3rd part denotes category 3 and gives the low-importance tasks to the VM-1 and VM-2. As has been listed in Table4, scheduling IoT Health-care (IoTH) tasks according to their significance with the MAX-MIN improves balance between ET, FT and WT for all of the tasks."

TABLE IV

THE MAX-MIN BASED ON THE SUGGESTED WORK

	tasks	lengths	VMs	WT	ET	FT
Low Importanc e Tasks	t2	70000	VM 1	0	141	141
	t13	20000	VM 2	0	41	41
	t11	2000	VM 2	40	4	44
	t4	1000	VM 2	48	2	50
	t10	1000	VM 2	50	2	52
Moder ate Import	t7	90000	VM 4	0	61	61
	t6	10000	VM 3	0	6.67	6.67
	t12	4000	VM 3	6.68	2.68	9.36
	t5	3000	VM 3	9.43	2	11.43
High Importance Tasks	t1	100000	VM 5	0	41	41
	t8	100000	VM 6	0	41	41
	t15	80000	VM 5	43	33	76
	t14	25000	VM 6	40	11	51
	t9	15000	VM 6	50	6	56
	t3	5000	VM 6	56	2	58

Table 5 shows decrease in AWT, AET, and AFT because the VMs first executed the large tasks for a long time while small and medium tasks were kept waiting to the point where large tasks were executed.

TABLE V

PERFORMANCE OF (MAX, MIN) WITH OR WITHOUT THE SUGGESTED WORK

	Max-Min algorithm	Max-Min based on proposed framework
AWT	34.445	32
AET	32.12	21.68
AFT	68.675	53.66

G. Research Methodology Motivations

1) First: Task scheduling using the existing tasks scheduling algorithms relies mainly on the task length, and this contributes to unfairness because large tasks are meant to wait for the smaller ones to be executed first as in the SJF, or medium or short tasks are kept waiting for larger tasks to be executed first as in (Max, Min).

2) Second: Many task scheduling algorithms have the problem of sequential task assignment to VMs without considering the available workload in every one of the VMs, the task length, or even the number of tasks assigned to every one of the VMs. So this accounts for the imbalance that leads to prolonged waiting time, execution time, and response time due to one VM being under-loaded while another VM is under-loaded or idle.

3) Third: The DC deals with a massive amount of the tasks from a variety of lengths and importance levels. However, it's unfair to consider only the lengths of tasks when assigning priority as seen in FCFS, (Max, Min), SJF, and PSO scheduling methods. So, the smallest and largest tasks may have similar priorities and importance in some instances. This requires a proper way of choosing a scheduling algorithm that can assign priority to tasks based on their importance, irrespective of length.

III. RESULTS AND DISCUSSION

The CloudSim simulation tool was used to implement the proposed scheduling method using the Eclipse IDE environment. The simulation considered some types of VMs and tasks of different lengths to ensure proper evaluation of the performance of the scheduling method.

A. Simulation Setup

The implementation and evaluation of the proposed scheme for IoTH tasks scheduling with the use of the TCVC based upon the task significance with the MAX-MIN algorithm of scheduling were presented in this section. The proposed method was evaluated and compared in terms of performance against other existing scheduling schemes, like the SJF, (Max, Min) and FCFS.

Setup:

1- Task Description: Tasks Length values={100,000, 70,000, 50,00, 1,000, 3,000, 10,000, 90,000, 100,000, 150,00, 1,000, 2,000, 4,000, 20,000, 25,000, 80,000}, Output Size=300 and File Size =300.

2- VM Descriptions: MIPS ={500, 500, 1,500, 1,500, 2,500, 2,500} RAM= 2,048, size of the Image =10,000, Number of the CPUs=1, BW= 1,000, vmm= "Xen"

3- Description of the Host: Storage= 100,000, RAM =2,048, MIPS=10,000 and BW= 10,000

4- DC Descriptions: String arch = "x 86"; // architecture of the system,

String vmm = "Xen";
String os = "Linux"; // OS,
double cost = 3; \\ The cost based on processing with this resource
double time zone = 10.01; \\ Time zone
double-cost (Per Mem) = 0.050; \\ cost of memory
double-cost Per Storage = 0.001; \\ cost of the storage

H. Impacts of suggested approach according to the Max, Min algorithm of scheduling

The AWT for the important tasks is shown in Fig.3 for different scheduling algorithms. Clearly, (Max, Min) with the suggested approach had performed better than (Max, Min) and FCFS methods without the proposed scheme, presenting an improvement of about 12 % over (Max, Min). Meanwhile, Figure 4 presents the AFT of different scheduling schemes for important tasks. (Max, Min) with the proposed work also performed better than SJF, (Max, Min) and FCFS methods, marking an achievement of about 24 % over (Max, Min).

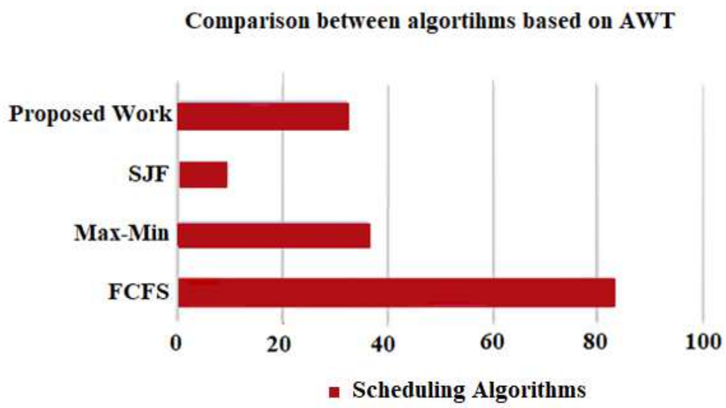

Fig. 3 AWT in the case of using (Max, Min) with the proposed task scheduling

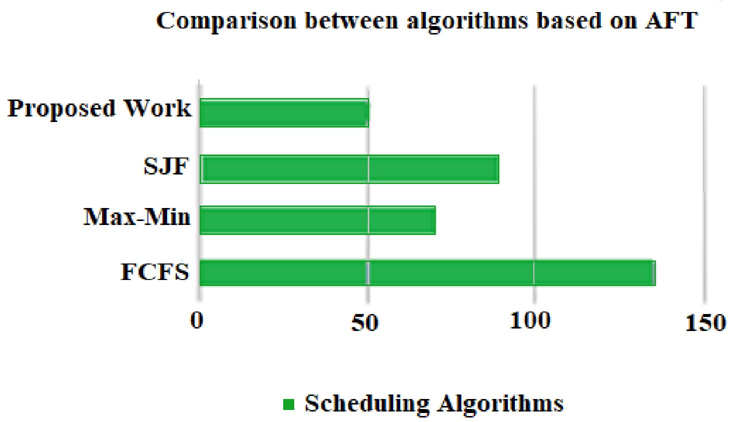

Fig. 4 AFT when using (Max, Min) with the proposed tasks scheduling

IV. Conclusion

In this research, the effort towards improving the IoTH task scheduling method performance in FC environment utilizing the new approach called TC, which uses tasks priority/significance. The application of the new approach with Max, Min (based on simulation studies) yielded better performance in terms of AET, AWT, fairness amongst tasks, AFT, and load balance among VMs in the VM list. Their approach also ensured real-time execution of the high priority (important) tasks generated by IoT devices and high efficiency with low latency. Hence, the proposed work is suitable of Real-Time (RT) remote patient monitoring. Note that the new approach ensured that VMs' capability must be proportional to the task's importance; important tasks are assigned to a class that requires high resource ability, while tasks of medium importance are assigned to classes with medium resource capability to ensure load balance and best performance.

REFERENCES

[1] M. Anuradha *et al.*, "IoT enabled cancer prediction system to enhance the authentication and security using cloud computing," *Microprocessors and Microsystems*, vol. 80, p. 103301, 2021.

[2] F. Muheidat and L. Tawalbeh, "Mobile and cloud computing security," in *Machine Intelligence and Big Data Analytics for Cybersecurity Applications*, Springer, 2021, pp. 461–483.

[3] G. Rekha, A. K. Tyagi, and N. Anuradha, "Integration of fog computing and internet of things: an useful overview," in *Proceedings of ICRIC 2019*, Springer, 2020, pp. 91–102.

[4] P. H. Vilela, J. J. P. C. Rodrigues, R. da R. Righi, S. Kozlov, and V. F. Rodrigues, "Looking at fog computing for e-health through the lens of deployment challenges and applications," *Sensors (Switzerland)*, vol. 20, no. 9. MDPI AG, May 01, 2020. doi: 10.3390/s20092553.

[5] P. H. Vilela, J. J. P. C. Rodrigues, R. da R. Righi, S. Kozlov, and V. F. Rodrigues, "Looking at fog computing for e-health through the lens of deployment challenges and applications," *Sensors*, vol. 20, no. 9, p. 2553, 2020.

[6] J. Zhao, P. Zeng, and K.-K. R. Choo, "An efficient access control scheme with outsourcing and attribute revocation for fog-enabled E-health," *IEEE Access*, vol. 9, pp. 13789–13799, 2021.

[7] Z. Bakhshi, G. Rodriguez-Navas, and H. Hansson, "Dependable fog computing: A systematic literature review," in *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2019, pp. 395–403.

[8] M. Sarrab and F. Alshohoumi, "Assisted Fog Computing Approach for Data Privacy Preservation in IoT-Based Healthcare," in *Security and Privacy Preserving for IoT and 5G Networks*, Springer, 2022, pp. 191–201.

[9] 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, 2020.

[10] I. S. B. M. Isa, T. E. H. El-Gorashi, M. O. I. Musa, and J. M. H. Elmirghani, "Energy Efficient Fog-Based Healthcare Monitoring Infrastructure," *IEEE Access*, vol. 8, pp. 197828–197852, 2020.

[11] M. Ahmed, R. Mumtaz, S. M. H. Zaidi, M. Hafeez, S. A. R. Zaidi, and M. Ahmad, "Distributed fog computing for Internet of Things (IOT) based ambient data processing and analysis," *Electronics (Basel)*, vol. 9, no. 11, p. 1756, 2020.

[12] S. Shukla, M. Hassan, D. C. Tran, R. Akbar, I. V. Paputungan, and M. K. Khan, "Improving latency in Internet-of-Things and cloud computing for real-time data transmission: a systematic literature review (SLR)," *Cluster Computing*, pp. 1–24, 2021.

[13] C. L. Stergiou, A. P. Plageras, K. E. Psannis, and B. B. Gupta, "Secure machine learning scenario from big data in cloud computing via internet of things network," in *Handbook of computer networks and cyber security*, Springer, 2020, pp. 525–554.

[14] X. Li, Y. Lu, X. Fu, and Y. Qi, "Building the Internet of Things platform for smart maternal healthcare services with wearable devices and cloud computing," *Future Generation Computer Systems*, vol. 118, pp. 282–296, 2021.

[15] M. Sriraghavendra, P. Chawla, H. Wu, S. S. Gill, and R. Buyya, "DoSP: A Deadline-Aware Dynamic Service Placement Algorithm for Workflow-Oriented IoT Applications in Fog-Cloud Computing Environments," in *Energy Conservation Solutions for Fog-Edge Computing Paradigms*, Springer, 2022, pp. 21–47.

[16] M. Sheikh Sofla, M. Haghi Kashani, E. Mahdipour, and R. Faghih Mirzaee, "Towards effective offloading mechanisms in fog computing," *Multimedia Tools and Applications*, vol. 81, no. 2, pp. 1997–2042, 2022.

[17] R. Priyadarshini, R. Kumar Barik, and H. Dubey, "Fog-SDN: A light mitigation scheme for DdoS attack in fog computing framework," *International Journal of Communication Systems*, vol. 33, no. 9, p. e4389, 2020.

[18] N. Premkumar and R. Santhosh, "Challenges and Issues of E-Health Applications in Cloud and Fog Computing Environment," in *Mobile Computing and Sustainable Informatics*, Springer, 2022, pp. 711–721.

[19] G. Fersi, "Study of middleware for Internet of healthcare things and their applications," in *International Conference on Smart Homes and Health Telematics*, 2020, pp. 223–231.

[20] A. R. Nair and S. Tanwar, "Fog Computing Architectures and Frameworks for Healthcare 4.0," in *Fog Computing for Healthcare 4.0 Environments*, Springer, 2021, pp. 55–78.

[21] V. C. M. Leung, X. Wang, A. Jamalipour, X. Chen, and S. Bouzefrane, "IEEE Access Special Section Editorial: Edge Computing and Networking for Ubiquitous AI," *IEEE Access*, vol. 9, pp. 90933–90936, 2021.

[22] J. Zhu, J. Hu, M. Zhang, Y. Chen, and S. Bi, "A fog computing model for implementing motion guide to visually impaired," *Simulation Modelling Practice and Theory*, vol. 101, p. 102015, 2020.

[23] T. Qayyum, Z. Trabelsi, A. W. Malik, and K. Hayawi, "Multi-level resource sharing framework using collaborative fog environment for smart cities," *IEEE access*, vol. 9, pp. 21859–21869, 2021.

[24] M. T. Hossain and E. Robson, "Cloudlet dwell time model and resource availability for vehicular fog computing," in *2021 IEEE/ACM 25th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, 2021, pp. 1–8.

[25] 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, 2022.

[26] A. Bozorgchenani, D. Tarchi, and W. Cerroni, "On-Demand Service Deployment Strategies for Fog-as-a-Service Scenarios," *IEEE Communications Letters*, vol. 25, no. 5, pp. 1500–1504, 2021.

[27] G. Ortiz, M. Zouai, O. Kazar, A. Garcia-de-Prado, and J. Boubeta-Puig, "Atmosphere: Context and situational-aware collaborative IoT architecture for edge-fog-cloud computing," *Computer Standards & Interfaces*, vol. 79, p. 103550, 2022.

[28] A. M. Farooqi, M. A. Alam, S. I. Hassan, and S. M. Idrees, "A Fog Computing Model for VANET to Reduce Latency and Delay Using 5G Network in Smart City Transportation," *Applied Sciences*, vol. 12, no. 4, p. 2083, 2022.

[29] J. Choi and S. Ahn, "Optimal Service Provisioning for the Scalable Fog/Edge Computing Environment," *Sensors*, vol. 21, no. 4, p. 1506, 2021.

[30] V. Dave and N. Joshi, "Fog computing enabled Ambient Assisted Healthcare systems," in 2019 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), 2019, pp. 1–7.

[31] Z. Ning *et al.*, "Mobile edge computing enabled 5G health monitoring for Internet of medical things: A decentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 463–478, 2020.

[32] T. Alam, M. Tajammul, and R. Gupta, "Towards the Sustainable Development of Smart Cities Through Cloud Computing," in *AI and IoT for Smart City Applications*, Springer, 2022, pp. 199–222.

[33] M. Bhatia, T. A. Ahanger, U. Tariq, and A. Ibrahim, "Cognitive intelligence in fog computing-inspired veterinary healthcare," *Computers & Electrical Engineering*, vol. 91, p. 107061, 2021.

[34] T. M. Fernández-Caramés, I. Froiz-Míguez, O. Blanco-Novoa, and P. Fraga-Lamas, "Enabling the internet of mobile crowdsourcing health things: A mobile fog computing, blockchain and IoT based continuous glucose monitoring system for diabetes mellitus research and care," *Sensors*, vol. 19, no. 15, p. 3319, 2019.

[35] T. M. Fernández-Caramés, I. Froiz-Míguez, O. Blanco-Novoa, and P. Fraga-Lamas, "Enabling the internet of mobile crowdsourcing health things: A mobile fog computing, blockchain and IoT based continuous glucose monitoring system for diabetes mellitus research and care," *Sensors*, vol. 19, no. 15, p. 3319, 2019.

[36] N. Tewari and S. K. Budhani, "Analysis of Risk and Security Within Fog Computing-Enabled e-Healthcare System in Uttarakhand," in *Cyber Security and Digital Forensics*, Springer, 2022, pp. 21–30.

[37] L. Lakshmi, A. N. Kalyani, G. N. Satish, D. Swapna, and M. P. Reddy, "The preeminence of Fog Computing and IoT enabled Cloud Systems in Health care," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2021, pp. 368–375.

[38] M. Abu-Elkheir, H. S. Hassanein, and S. M. A. Oteafy, "Enhancing emergency response systems through leveraging crowdsensing and heterogeneous data," in *2016 international wireless communications and mobile computing conference (IWCMC)*, 2016, pp. 188–193.

[39] R. Mahmud and A. N. Toosi, "Con-pi: A distributed container-based edge and fog computing framework," *IEEE Internet of Things Journal*, 2021.

[40] A. Mukherjee, S. Ghosh, A. Behere, S. K. Ghosh, and R. Buyya, "Internet of health things (IoHT) for personalized health care using

integrated edge-fog-cloud network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 943–959, 2021.
[41] P. G. Shynu, V. G. Menon, R. L. Kumar, S. Kadry, and Y. Nam, "Blockchain-based secure healthcare application for diabetic-cardio disease prediction in fog computing," *IEEE Access*, vol. 9, pp. 45706–45720, 2021.
[42] R. Dhanagopal and B. Muthukumar, "IOT Based Energy Efficient Early Landslide Detection System," *Journal of Electrical Engineering*, vol. 21, no. 1, p. 10, 2021.
[43] D. Sarabia-Jácome, R. Usach, C. E. Palau, and M. Esteve, "Highly-efficient fog-based deep learning AAL fall detection system," *Internet of Things*, vol. 11, p. 100185, 2020.