International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

June 2022	Issue 51	Volume 14	Number 2	Pages 176-182

EXTENDED HYBRID CLUSTER ALGORITHM FOR COMPUTATION OFFLOADING IN FOG COMPUTING

A.K. Jha [1] M.P. Patel [2] T.D. Pawar [3]

1. Gujarat Technological University, Chandkheda, Ahmedabad, India, ashwinibjha@gmail.com
2. Computer Engineering Department, Devang Patel Institute of Advance Technology and Research, CHARUSAT, Anand, India, mppatel.adit@gmail.com
3. Electronics Department, Birla Vishvakarma Mahavidyalaya, Anand, India, tanmaypawar@gmail.com

Abstract- Technological advances have taken place at a whirlwind pace over the past few decades, new devices have flooded functionalities in a way that technology has built a base in almost every arena of human existence. Integrated microdevices for smart grid, smart health, smart parking, and smart farming generate a vast amount of data to integrate, analyze, process, and store, computational data. These data cannot be handled natively, it necessitates the installation of a fog server of reasonable capacity for time-critical applications. Various factors can affect the decision about offloading tasks from the host to the fog network. The proposed research simulates fog offloading using iFogSim. Fog nodes' processing capacity and the threshold value are calculated after considering the CPU length, RAM, uplink-downlink bandwidth, power consumption, and rate per million instructions per second (MIPS). By examining the state of each node in the fog network, the proposed algorithm determines whether to offload a task to a fog network or cloud server. The research provides a unique approach of computation offloading. Additionally, the algorithm ensures that all fragments of tasks should be allocated to a single device to reduce the communication time between devices.

Keywords: Fog Computing, Cloud Computing, Extended Hybrid Algorithm, Computation Offloading, Smart Surveillance, Edge Computing.

1. INTRODUCTION

Cloud is the service that enables a lot of things to happen on the internet. IT departments can focus on their businesses and projects instead of just maintaining and managing their data centers. Cloud technology has some downsides, especially when it comes to the IoT services, such as raised latency where cloud-based IoT applications demand reduced latency, however, because of the distance between clients and data centers [1], the cloud cannot guarantee it, and the technical problems and disruptions in networks can happen in some IP-enabled machines and cause clients to experience network unavailability [2].

As thousands of people share data over a network, there is a chance that someone might get unauthorized access to the data and might lose it [3]. The problem can be partly fixed by using fog networks.

A fog infrastructure is distributed and decentralized, as shown in Figure 1 in contrast to a cloud infrastructure that is centralized. An implementation of fogging brings numerous advantages to the IoT enabled systems, Big Data analytics, and real-time analytics [3]. A significant benefit of fog computing is its low latency as it is geographically nearer to the user device [4], and it can reply promptly, the bandwidth cannot be an issue as instead of sending the pieces of information together via a single channel, they are aggregated at different locations. It is very important and crucial for a network to be able to collect and process data through the cloud and edge devices, and computation offloading is one of the most vital tools available to deal with the information overload we currently face. Healthcare solutions, autonomous vehicles, and other time-critical applications require the computing machines to be located nearby the user device.

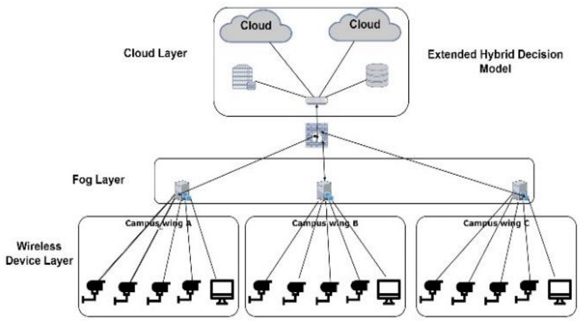

Figure 1. Extended hybrid decision model architecture

Consequently, a fog topology was created, and the computing was delegated to the fog server via an external network to improve processing capacity and address hardware constraints [5], such as storage and processing power, and it can complete the task more quickly and efficiently.

There are six sections in the proposed paper: Section 2 provides an outline of the related work; Section 3 defines proposed system architecture; Section 4 details the experimental setup; Section 5 enumerates experimental outcomes; and lastly, Section 6 concludes the research and gives future course.

2. LITERATURE REVIEW

Cloud computing can store, process, and manage data for an extended period, but often accessing the cloud layer to communicate and retrieve data raises network time and uses elevated network bandwidth because of that rest of application are affected [6]. Fog nodes reduce latency because they bypass frequent cloud accesses, it represents a dispersed network concept that is strongly related to cloud computing and demonstrates the concept of two different environments connected through a network. First introduced by Cisco in 2005, this new way of computing enables enterprise networks to widen to the edge of the cloud [7]. Fog environments enumerates host, Fog server, and Cloud [8].

It is the IoT layer's responsibility to collect and send data produced from devices to the Fog layer. Between the cloud and the Internet of Things, the fog layer serves as a link. This layer contains fog devices that handle stored data and transmit the solutions to the cloud server to be stored for forthcoming references [9]. By contributing their idle resources to a Fog environment, individuals or organizations can process applications within a Fog environment. A provider should deliver resources in a manner that benefits both the provider and the user by charging based on usage [10]. Among the few initiatives undertaken to advance fog computing, Open Fog Consortium is one of them, but the exact definition of fog is not agreed upon, and it may be described as a source of decentralized computing power that allows remote computing services to be stretched to the edge of the network.

Recent years have seen an increased interest in the topic of offloading in mobile computing. The study of Deng, et al. [11] aimed to determine the effect of allocating the load on fog devices on communication delay and power utilization. In [12], the authors highlight fog computing's benefits, concluding that a fog device placed between the cloud and end-users reduces network traffic by 90%. In [13], performance improvement and energy savings were identified as two key factors to justify offloading tasks. For task scheduling among clouds, fogs, and edges, LPF (Least Processing Cost First) [14] has been used to achieve ideal execution time and optimal network use. According to [15], a novel offloading approach led to a significant decrease in the overall system's execution time and energy consumption.

In [16], an energy-aware distribution strategy was suggested, and the simulation results showed that the cloud only configuration consumed 2.72 percent more energy than the fog configuration. A simulation was conducted in [17] and [18] to check how mobility support impacts offloading policy. Depending on many factors, like processing power, latency requirements, turnaround time, deadlines, transmission capacity, power limitations and execution cost, offloading policy may differ. Instead of simulating and analyzing proposed methodologies, most of the work has relied on mathematical models. The simulations presented in [16], [19-22] attempt to optimize the various performance parameters. Although all of them has a focus on just one performance parameter, [23] is the first study to incorporate multiple performance parameters at the same time.

In [20], Awaisi, et al., discuss efficient parking systems using the iFogSim model to minimize latency and network usage. In their study, they compare cloud-only deployments with fog-enabled deployments. However, there is a serious flaw in the system specifications: the communication bandwidth provided to the system differs significantly. As a result, fog-based computation offloading performs better than cloud-based deployment, and the results are strikingly different. Thus, a cloud-based description is inaccurate and is not appropriate for implementation.

Most of the policies under the literature review do not correlate since they work under different constraints, employ different testing scenarios, and use distinct simulators. Moreover, there are no guidelines for fog architectures, so comparing the various policies is highly difficult. Fog computing does not have an established protocol, although the Open Fog Consortium anticipates Open Fog Architecture [24] as the standard for developing fog-enabled architectures.

3. PROPOSED SYSTEM ARCHITECTURE

The recommended design is comprised of three layers, as illustrated in Figure 1. Cameras above the observation slots are in the first layer and are accountable for taking image data of the area and verifying whether it is crowded or not. Fog nodes are attached to the smart cameras via a microcontroller unit in the following layer of the structure [25]. There is a third tier that comprises of a cloud connected to fog server that manages and stores the data associated with images for longer periods of time. Using LEDs installed around the area, the LED screen will display a message showing whether people are maintaining safe distances from each other as shown in Figure 1. Also, the automatic system [26] will be helpful for authorities to determine whether the public around them is maintaining distance protocol. Surveillance smart cameras are positioned in multiple locations to encircle a larger area.

This architecture uses a microcontroller to act as a link in the middle of the fog nodes and cameras [27]. Fog devices communicate with the cloud and transmit tasks to it on a periodic basis. The LED alerts the administrative control office about people violating the set distance protocol if the LED indicates that there has been an infringement for a defined period, the administrative control office can act against the violators.

Fog nodes are placed in every surveillance area and connected to a central cloud server, The latency and network usage of every fog node is assumed to be same, but if the data is uploaded and retrieved from a consolidated cloud server, the upload time and the network usage will increase till a limit, but if the count of fog servers are not increased and the topology experiences a slag in execution then the extended hybrid cluster algorithm will ensure to transfer the computational data to the cloud. In cloud and fog computing, clustering is an effective way to disseminate data [28], [29]. In the following sections, as shown in Figure 1 and Figure 2 we examine the workings of the projected architecture.

3.1. Camera Layer

Tier-1 of the smart surveillance system consists of the microcontroller chip and cameras. The image processing approach of the camera was preferable for identifying and surveying the region based on distance between people [28]. In this exercise, we assess the latency and network utilization of fog with cloud. The chip is designed to transfer the images of surveillance captured by the camera to the fog node, where they can be further processed. As soon as the image procurement stage is completed, the RGB image segmentation is performed, afterwards, the RGB data is converted to grey scale data. Greyscale images are optimized by threshold techniques [30], followed by the acquisition of a binary image for segmentation.

3.2. Fog Layer

Tier-2 between the camera and the cloud, it is the middle layer [31]. The microcontroller gathers the image data through the smart cameras and to monitor the surveillance area's status, smart LEDs at the venue and authority office display the status of the region based on the update from the fog. Before data is transferred to the cloud, it is stored on the fog node for a set length of time. The fog computing infrastructure [32] provides a layer between fog nodes and the cloud that can be used to execute, analyze, and process the concurrent data of the surveillance region.

3.3. Cloud Layer

Tier-3 in the proposed framework the clouds have the role of analyzing the image data when the fog node is heavily affected by the workload and the proxy servers are used to enable interaction between fog and cloud servers [11]. In addition to many important traits of fog computing, interoperability plays a large role when considering how diverse edge nodes really are [33]. In fog nodes, resources are shared among them so that adjoining fog device can satisfy its computing and memory demands [34]. In our scenario, we assume fog nodes can exchange crucial information with neighboring nodes through a proxy server. Communication between fog nodes will not be impacted by latency, which is an inherent characteristic of interoperability [33]. And in case fog layer is extremely burdened then the extended hybrid cluster algorithm will easily transfer the data to cloud as explained in Algorithm 1.

The Extended Hybrid cluster algorithm attempts to locate several components on one computing node so that overhead of intercommunication between the various application modules will be reduced [35] and the energy can be saved [33]. The goal of the. Computation offloading extended hybrid cluster algorithm is to reduce radio band utilization [36]. Fog devices comprise the network topology, and they have been arranged randomly in levels, based on the associations with other Fog nodes. The computational power of Fog devices is expected to increase as we step up in the network architecture.

Algorithm 1. Computation Offloading using Extended Hybrid Cluster Algorithm

```
Input feed: File of User App task, Fog servers, and Cloud server
Output feed: Assigning user task to fog or cloud server
For R ∈ ROUTES do across all network routes
Task_Allocated= {}
load=0
Threshold_Count=N
for task T ∈ User_Application does (T=T1+T2+. . . +Tn)
T1 is placed on F ∈ R
if CPU_Tn^req ≤ CPU_F^Thres then
Place Tn on device F
Add T to Allocated
Break
end if
load=load+ CPU_T^req
end for
for task T ∈ User_Application do
if T not placed on F ∈ R and CPU_F^avail ≤ CPU_T^Thres
Place T1 on Device C
as CPU_C^avail ≈ ∞
Keep Adding Tn to allocated
Break
End if
load=load+ CPU_T^req
end for
end for
```

There are user nodes U, fog server F, and cloud server C. Each user node will have various application modules termed as Task T. All the fog and cloud routes are termed as R. If a user decides to offload its task to a fog server, then all the tasks of the user application should offload its task only to that single device based upon the uniquely derived threshold value, else it should offload to cloud server for final execution as shown in the Figure 2 topology of smart surveillance cameras connected to fog nodes and cloud nodes through proxy. It is assumed that the cloud has infinite processing power [37]. A task allocated list is created to keep a count of the total allocated task. A load variable is initialized to calculate the load exerted on the computing system.

4. EXPERIMENTAL SETUP

As part of our simulation, high-definition smart cameras are used for taking images of the surveillance area to determine the presence of crowd and to recognize faces [38]. The images are later sent to the cloud or fog node to process and detect the status of the region and display the result on LED screen attached to the fog server via a network medium. Fog nodes and cloud servers are connected by proxy servers. A proxy server connects fog to the cloud server. Every region was given at least one fog node.

Using the simulation environment, we considered the cameras as sensors since they are smart and linked via a microcontroller, as described in [39]. A larger count of cameras was added to test the results for different situations. Our study compared the effect of adding more cameras in a fog node on the amount of latency and overall network utilization. Figure 2 illustrates the topology designed by iFogSim for the evaluation of fog setup findings.

Table 1. Parameter configuration of the various devices

Parameter	Fog	Proxy	Cloud
CPU (MIPS)	2000	3000	50000
RAM (MIPS)	4000	8000	40000
Up Bandwidth (MB)	5000	5000	5000
Down Bandwidth (MB)	12000	12000	12000
Device Level	2	1	0
Rate/(MIPS)	0.0	0.04	0.01

With an increased number of surveillance regions, fog nodes are formed, and they become responsible for determining the crowd status. Fog servers' latency and network usage will increase as the number of cameras and screens increases. Processing is performed at fog nodes, meaning that the cloud is not burdened with computation, so the computation burden is reduced significantly. When cameras are directly connected to a cloud server via a router, latency and bandwidth consumption are higher. As long as the same regions have their respective proxy servers connected to the cloud via a router, network latency does not become an issue.

There is an assumption that the cloud has infinite processing power [37] and any task assigned to it will be processed in record minimum time. Table 1 related to simulation of a scenario based on fog and cloud servers presents several parameters related to setting up cloud, proxy, and fog servers.

Table 2. Network latency between devices

Device-Device		Latency
Sensor	Edge	1
Edge	Actuator	1
Edge	Proxy	5
Proxy	Cloud	100

In addition to processing ability and RAM, configuration parameters include bandwidth, rate, and cost of million instruction per second (MIPS), busy power, and idle power. As part of a cloud-based performance evaluation scenario, several cameras and LEDs are connected to the router for performance monitoring. In Table 2, network latency incurred during communication between various devices are mentioned. Simulated images were taken continuously by using smart cameras in each surveillance region while the cloud and fog were switched in the same networking configuration and varying computational data.

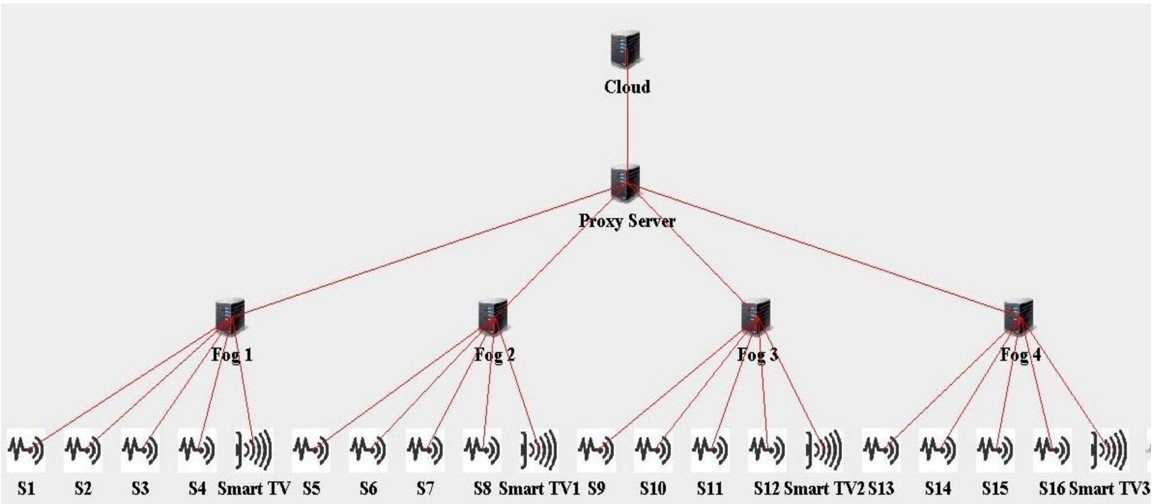

Figure 2. A topology of smart surveillance cameras, fog servers, proxy servers, and cloud servers

4.1. Experimental Results and Comparisons

In this part, we are attempting to compare the architecture at two levels. Initially, we present results relating to the execution time in the proposed fog computing architecture, and then we compare them with cloud computing implementations [40]. At the second level, we will compare our proposed model with that of a smart car parking system [28].

The number of data nodes in the proposed model was increased after every simulation. Through the addition of eight regions and eight cameras per surveillance area, we have reached 64 smart cameras in the fog network.

The same number of cameras was again put into the cloud application, and the results of the comparison showed that the cloud appeared to have a slight advantage over the fog in the dense application. Comparing the fog-based implementation to the cloud-based one, experiments show that fog-based implementations achieve a low latency and a low execution time in the beginning. As shown in Figure 3, there came a point where there were 64 smart devices, while the execution time in the fog server kept slanting. In contrast, the cloud server, which has a vastly higher computational capacity, significantly decreases the execution time when compared with a fog server.

Table 3. Cloud and fog execution latency using hybrid cluster algorithm

Smart Cameras	Cloud Latency	Fog Latency
4	667	528
8	1076	835
16	1551	1277
32	2483	2136
40	2809	2588
48	3117	2835
64	3950	3858
72	4063	4380
80	4195	4765
88	4505	5078
96	4908	5510
104	5199	5986

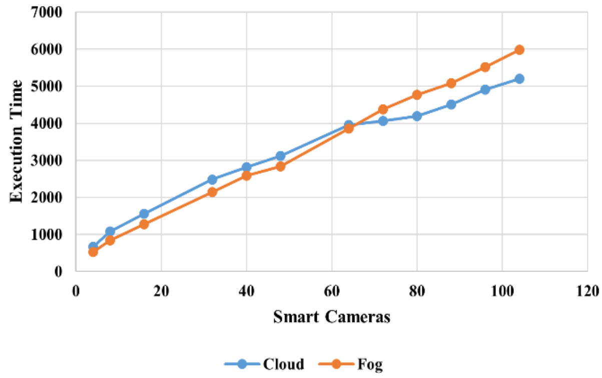

Figure 3. Computation offloading execution comparison for cloud and fog server

A total of 24 simulations were performed, and the number of cameras varied from four cameras per region to 104 cameras spread over 13 regions each with eight cameras as shown in table 3. As shown in Figure 3, cloud offloading execution times have stabilized compared to fog nodes. Based on the payload tuple type and network tuple length, it is proposed to calculate the threshold [41] value of the fog server in the extended hybrid cluster algorithm. When the number of devices increases, the computation time on the fog server [42] increases until it reaches the threshold value, after which the cloud server [43] is switched to offload the computation. In Figure 3, the execution time graph of the fog-based execution is shown alongside the results of cloud-based execution.

The second comparative scenario is based on a smart parking system [28]. Fog computing is primarily used to minimize latency and network usage. When using iFogSim for minimizing latency and network usage, the results were compared to cloud-enabled deployments. Figure 4 illustrates the significant reduction in latency associated with fog-based implementations. It must be noted, however, that this system has a serious flaw, as can be seen by the graph in Figure 4 which rises dramatically after 42 devices are added. Cloud-only approaches have a bandwidth of 100 megabytes, while fog-based approaches have a bandwidth of 1000 megabytes, while both utilize wireless as the transmission medium. Therefore, the effectiveness of the parking system cannot be determined, which eliminates the possibility of comparative analysis.

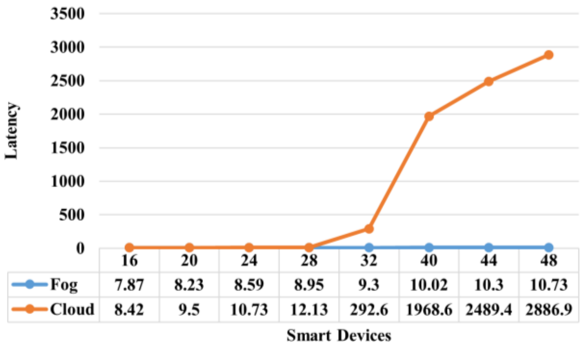

Figure 4. Execution Latency comparing cloud and fog

5. CONCLUSION AND FUTURE WORK

Fog computing has recently gained importance, especially in time critical applications. Increasing data generation devices have also increased the demand for faster responses. We therefore presented a fog computing based smart crowd surveillance system that uses computer vision practice to identify the status of the region, allowing the administrative authorities to administer a region efficiently in minimum time. Comparing the proposed fog-based architecture to a cloud-based model till a threshold level. The results show that not only does fog minimize latency, but its network usage is also lower. With more cameras, the network traffic increases, and additional strain is put on the server. Consequently, the threshold factor of the extended hybrid cluster algorithm switches automatically to cloud mode for computation without affecting the topology's normal operation. As a result, we can seamlessly and automatically offload from fog servers to cloud servers and vice versa.

The use of cameras for crowd surveillance is a limitation of the research because the images are being stored in the cloud, this may cause privacy issues for visitors. Nevertheless, it is crucial to safeguard the confidentiality of cloud files by applying appropriate encoding methods, and this will be a capable future research objective. Furthermore, it is also imperative to point out that with the increased number of surveillance areas, the massive implementation of the projected structure will require further load balancing on fog computing technology to provide competence.

REFERENCES

[1] M. Alojail, M. Bhatia, S. Sharma, S. Bhatia, "Fog Computing Mitigate Limitations of Cloud Computing", International Journal of Recent Technology and Engineering, Vol. 8, pp. 1732-1736, January 2020.
[2] B.K. Mohanta, D. Jena, S.S. Panda, D. Gountia, "Decentralized Secure Fog Computing in Cloud-Fog-IoT Infrastructure Using Blockchain", International Conference on Emerging Technologies of Information and Communications, p. 6, March 2019.
[3] M. Patel, A. Pandya, "Edge Computing: Design a Framework for Monitoring Performance between Datacenters and Devices of Edge Networks", International Journal of Computer and Mathematical Sciences, Vol. 6, No. 6, pp. 73-77, 2017.

[4] H.F. Atlam, R.J. Walters, G.B. Wills, "Fog Computing and the Internet of Things: A Review", Big Data and Cognitive Computing, Vol. 2, No. 2, June 2018.
[5] R.A.C. da Silva, N.L.S. da Fonseca, "On the Location of Fog Nodes in Fog-Cloud Infrastructures", Sensors, Vol. 19, No. 11, January 2019.
[6] E. Badidi, E. Sabir, "Fog Computing for Smart Cities' Big Data Management and Analytics: A Review", Future Internet, Vol. 12, p. 190, October 2020.
[7] N. Jagani, P. Jagani, S. Shah, "Big Data in Cloud Computing: A Literature Review", International Journal of Engineering Applied Sciences and Technology, Vol. 5, March 2021.
[8] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, "Fog Computing and its Role in the Internet of Things", The MCC Workshop on Mobile Cloud Computing, Ed. 1, pp. 13-16, New York, USA, August 2012.
[9] I. Ud Din, et al., "The Internet of Things: A Review of Enabled Technologies and Future Challenges", IEEE Access, Vol. 99, p. 1, December 2018.
[10] A. Mijuskovic, A. Chiumento, R. Bemthuis, A. Aldea, P. Havinga, "Resource Management Techniques for Cloud/Fog and Edge Computing: An Evaluation Framework and Classification", Sensors, Vol. 21, No. 5, January 2021.
[11] R. Deng, R. Lu, C. Lai, T. Luan, H. Liang, "Optimal Workload Allocation in Fog-Cloud Computing Towards Balanced Delay and Power Consumption", IEEE Internet of Things Journal, p. 1, January 2016.
[12] M. Patel, A. Mehta, N. Chauhan, "Design of Smart Dashboard Based on IoT & Fog Computing for Smart Cities", The 5th International Conference on Trends in Electronics and Informatics (ICOEI), 2021.
[13] S. Battula, S. Garg, R. Naha, P. Thulasiraman, R. Thulasiram, "A Micro-Level Compensation-Based Cost Model for Resource Allocation in a Fog Environment", Sensors, Vol. 19, pp. 1-21, July 2019.
[14] B. Varghese, N. Wang, D. Nikolopoulos, R. Buyya, "Feasibility of Fog Computing", Handbook of Integration of Cloud Computing, Cyber Physical Systems and Internet of Things, R. Ranjan, K. Mitra, P. Prakash Jayaraman, L. Wang, A. Zomaya, (Eds.), Springer, 2019.
[15] M. Patel, N. Chauhan, "Smart Dashboard: A Novel Approach for Sustainable Development of Smart Cities using Fog Computing", The 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019.
[16] K. Kumar, J. Liu, Y.H. Lu, B. Bhargava, "A Survey of Computation Offloading for Mobile Systems", Mobile Net. Appl., Vol. 18, No. 1, February 2013.
[17] N. Mohan, J. Kangasharju, "Edge-Fog Cloud: A Distributed Cloud for Internet of Things Computations", Cloudification of the Internet of Things (CIoT), pp. 1-6, November 2016.
[18] Q. Zhu, B. Si, F. Yang, Y. Ma, "Task Offloading Decision in Fog Computing System", China Communications, Vol. 14, No. 11, November 2017.
[19] M. Mahmoud, J. Rodrigues, K. Saleem, J. Al Muhtadi, N. Kumar, V. Korotaevf, "Towards Energy-Aware Fog-Enabled Cloud of Things for Healthcare", Computers and Electrical Engineering, Vol. 67, March 2018.
[20] M.M. Lopes, W.A. Higashino, M.A.M. Capretz, L.F. Bittencourt, "MyiFogSim: A Simulator for Virtual Machine Migration in Fog Computing", The 10th International Conference on Utility and Cloud Computing, pp. 47-52, New York, USA, December 2017.
[21] M. Taneja, A. Davy, "Resource Aware Placement of IoT Application Modules in Fog-Cloud Computing Paradigm", The IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 1222-1228, May 2017.
[22] L.F. Bittencourt, J. Diaz Montes, R. Buyya, O.F. Rana, M. Parashar, "Mobility Aware Application Scheduling in Fog Computing", IEEE Cloud Computing, Vol. 4, No. 2, March 2017.
[23] M.I. Naas, J. Boukhobza, P. Raipin Parvedy, L. Lemarchand, "An Extension to iFogSim to Enable the Design of Data Placement Strategies", The 2nd International Conference on Fog and Edge Computing (ICFEC), pp. 1-8, May 2018.
[24] F. Rahman, T.W. Au, S.H. Newaz, W. Suhaili, "A Performance Study of High-End Fog and Fog Cluster in iFogSim", Computational Intelligence in Information Systems Conference (CIIS 2018), pp. 87-96, 2019.
[25] Z. Rezazadeh, D. Rahbari, M. Nickray, "Optimized Module Placement in IoT Applications Based on Fog Computing", The Iranian Conference on Electrical Engineering (ICEE), Sadjad University of Technology, Mashhad, Iran, pp. 1553-1558, May 2018.
[26] S.S. Gill, P. Garraghan, R. Buyya, "ROUTER: Fog Enabled Cloud based Intelligent Resource Management Approach for Smart Home IoT Devices", Journal of Systems and Software, Vol. 154, pp. 125-138, August 2019.
[27] J.P.D. Clerck, "The Open Fog Reference Architecture and the IEEE 1934 Standard", The i-SCOOP, June 30, 2018, www.i-scoop.eu/openfog-reference-architecture-ieee-1934-standard (March 21, 2022).
[28] K.S. Awaisi, et al., "Towards a Fog Enabled Efficient Car Parking Architecture", IEEE Access, Vol. 7, pp. 159100-159111, 2019.
[29] S. Ouahou, S. Bah, Z. Bakkoury, "New Machine Learning Solution Based on Clustering for Delay-Sensitive Application in Vanet", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 49, Vol. 13, No. 4, pp. 7-16, December 2021.
[30] M. Aazam, E.N. Huh, "E-HAMC: Leveraging Fog Computing for Emergency Alert Service", International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), pp. 518-523, March 2015.
[31] I. Sarkar S. Kumar, "Fog Computing Based Intelligent Security Surveillance Using PTZ Controller Camera", The 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-5, July 2019.
[32] A. Rahman, A. Salam, M. Islam, "An Image Based Approach to Compute Object Distance", International

Journal of Computational Intelligence Systems, Vol. 1, pp. 304-312, March 2012.
[33] H. Barati, A. Movaghar, A. Rahmani, A. Sarmast, "A Distributed Energy Aware Clustering Approach for Large Scale Wireless Sensor Network", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 13, Vol. 4, No. 4, pp. 125-132, December 2012.
[34] H. Sabireen V. Neelanarayanan, "A Review on Fog Computing: Architecture, Fog with IoT, Algorithms and Research Challenges", ICT Express, Vol. 7, Issue 2, pp. 162-176, June 2021.
[35] A.V. Dastjerdi, H. Gupta, R.N. Calheiros, S.K. Ghosh, R. Buyya, "Fog Computing: Principles, Architectures, and Applications", Internet of Things, Elsevier, pp. 61-75, 2016.
[36] R. Mahmud, F.L. Koch, R. Buyya, "Cloud Fog Interoperability in IoT-enabled Healthcare Solutions", The 19th International Conference on Distributed Computing and Networking, Vol. 32, Issue 10, pp. 1-32, Varanasi, India, 2018.
[37] Z. Zhou, H. Liao, B. Gu, S. Mumtaz, J. Rodriguez, "Resource Sharing and Task Offloading in IoT Fog Computing: A Contract-Learning Approach", IEEE Transactions on Emerging Topics in Computational Intelligence, Vol. PP, pp. 1-14, October 2019.
[38] L. Shooshtarian, D. Lan, A. Taherkordi, "A Clustering-Based Approach to Efficient Resource Allocation in Fog Computing", Pervasive Systems, Algorithms and Networks, pp. 207-224, 2019.
[39] F. Sufyan, A. Banerjee, "Computation Offloading for Smart Devices in Fog-Cloud Queuing System", IETE Journal of Research, pp. 1-13, January 2021.
[40] M.R. Mesbahi, A.M. Rahmani, M. Hosseinzadeh, "Reliability and High Availability in Cloud Computing Environments: A Reference Roadmap", Hum. Cent. Comput. Inf. Sci., Vol. 8, No. 1, Art. No. 1, December 2018.
[41] M. Al Sa'd, S. Kiranyaz, I. Ahmad, C. Sundell, M. Vakkuri, M. Gabbouj, "A Social Distance Estimation and Crowd Monitoring System for Surveillance Cameras", Sensors, Vol. 22, No. 2, Art. No. 2, January 2022.
[42] H. Gupta, A. Vahid Dastjerdi, S.K. Ghosh, R. Buyya, "iFogSim: A toolkit for modeling and simulation of Resource Management Techniques in the Internet of Things, Edge and Fog Computing Environments", Software: Practice and Experience, Vol. 47, No. 9, pp. 1275-1296, 2017.
[43] G. Mondragon Ruiz, A. Tenorio Trigoso, M. Castillo Cara, B. Caminero, C. Carrion, "An Experimental Study of Fog and Cloud Computing in CEP-Based Real-Time IoT Applications", Journal of Cloud Computing, Vol. 10, No. 1, June 2021.

BIOGRAPHIES

Ashwini Kumar Jha was born in Sitamarhi, India in 1987. He received his B.E. and M.E. degrees in Computer Engineering from Saurashtra University Gujarat, India, and St. Peters University Chennai, India, in 2008 and 2011, respectively. Currently, he is pursuing Ph.D. at Gujarat Technological University, India. He is employed as an Assistant Professor at Madhuben and Bhanubhai Patel Institute of Technology (MBIT), CVM University, India. His research interests are fog computing, computation offloading, and wireless networking. He has supervised over 50 projects at the undergraduate level. He has published 5 articles in different journals and conferences.

Minal Patel was born in Vadodara, India in 1980. He received the B.E. and M.E. degree from South Gujarat University (currently, Veer Narmad University), Gujarat, India and Sardar Patel University, Gujarat, India, and Ph.D. degree in Computer Engineering from Dharmsinh Desai University, Nadiad (Gujarat, India) in 2002, 2007 and 2018 respectively. He is working as Assistant Professor in Computer Engineering Department at Devang Patel Institute of Advance Technology and Research (DEPSTAR), Charotar University of Science and Technology (CHARUSAT), India. His areas of interest are computing (cloud/fog computing), machine learning and security (network and blockchain security). He had supervised two Master of computer Engineering thesis and three Ph.D. scholars have been working under his supervision in blockchain technology, fog computing/cloud computing and machine learning. He guided 25+ projects in B.E. level. He has total 35 publications in different conferences and journals and 10 papers are in Scopus Indexed conferences/journals.

Tanmay D. Pawar was born in India, 1976. He completed his B.E. degree in Electronics Engineering at DDIT, Nadiad, and his MTech., and Ph.D. degrees in Electronics Engineering at IIT Bombay, Mumbai, India in 1996, 2002, and 2008, respectively. Currently, he is a Professor of Electronics Engineering at Birla Vishvakarma Maha Vidyalaya, Vallabh Vidyanagar, Gujarat, India. He is also the Board of Studies member of Electronics and Communication Engineering and allied branches at the Gujarat Technological University, Ahmedabad, India. He teaches courses like digital signal processing, probability theory, and stochastic processes. his research interests are in digital signal processing and patten classification.