

## Research Article

# Design of Resource-Aware Load Allocation for Heterogeneous Fog Computing Environments

Syed Rizwan Hassan <sup>1</sup>, Ishtiaq Ahmad <sup>1</sup>, Ateeq Ur Rehman <sup>2,3</sup>, Seada Hussien <sup>4</sup>,  
and Habib Hamam<sup>3,5,6,7</sup>

<sup>1</sup>Department of Electrical Engineering, The University of Lahore, Lahore 54000, Pakistan

<sup>2</sup>Department of Electrical Engineering, Government College University, Lahore 54000, Pakistan

<sup>3</sup>Faculty of Engineering, Uni de Moncton, Moncton, NB E1A 3E9, Canada

<sup>4</sup>School of Electrical and Computer Engineering, Haramaya Institute of Technology, 138 Dire Dawa, Ethiopia

<sup>5</sup>International Institute of Technology and Management, Libreville BP1989, Gabon

<sup>6</sup>Spectrum of Knowledge Production & Skills Development, Sfax 3027, Tunisia

<sup>7</sup>Department of Electrical and Electronic Engineering Science, School of Electrical Engineering, University of Johannesburg, Johannesburg 2006, South Africa

Correspondence should be addressed to Syed Rizwan Hassan; syedrizzwanhassan@gmail.com, Ateeq Ur Rehman; ateeq.rehman@gcu.edu.pk, and Seada Hussien; seada.hussien@aastu.edu.et

Received 2 February 2022; Revised 24 April 2022; Accepted 28 April 2022; Published 7 June 2022

Academic Editor: Maode Ma

Copyright © 2022 Syed Rizwan Hassan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The architecture employed by most of the researchers for the deployment of latency-sensitive Internet of Things (IoT) applications is fog computing. Fog computing architecture offers less delay as compared to the cloud computing paradigm by providing resource constraint fog devices close to the edge of the network. Fog nodes process the incoming data by utilizing available resources which reduces the volume of data to be sent to the cloud server. Fog devices having dissimilar processing capabilities are present in a system. The connection of suitable sensor nodes to the parent fog node plays an essential role in achieving the optimum performance of the system. In this paper, we have designed an algorithm that dynamically assigns appropriate sensor devices to fog nodes to achieve a reduction in network utilization and latency. The proposed algorithm estimates the volume of information detected by an edge device from the rate of sensing frequency of the sensor attached to the edge device. The proposed policy while connecting the network nodes takes into account the heterogeneity and processing capability of the devices. Several evaluations are performed on multiple scales for the evaluation of the proposed algorithm. The outcomes of the evaluations confirm the effectiveness of the proposed algorithm in achieving a reduction in network consumption and end-to-end delay.

## 1. Introduction

Devices under the everyday use of human beings are connected to the Internet owing to the Internet of Things (IoT) technology. Management of the big data generated by these devices needs high processing and storage units. The growing demand of IoT applications results in a huge increase in interconnection and deployment of IoT devices. The number of IoT devices to be present in the system by 2025 is expected to be 1 trillion [1], whose

financial impression will be nearly equivalent to 11% of the world economy.

A large number of applications of diverse kinds of nature are implemented using cloud and fog computing architectures. The applications implemented on the cloud paradigm consist of sensor nodes that contain sensors that are used to detect the environment. The effectiveness of the results deduced from the sensed data depends on the frequency and quality of the information sensed by the sensor nodes. In conventional cloud-based architecture, the sensors are

directly linked to the cloud server. The data sensed by the sensor nodes are transmitted directly to the cloud for further processing [2]. Cloud computing provides a centralized approach for the implementation of IoT applications. In cloud architecture, a resourceful cloud server is located in a centralized way where all the data generated by the end devices is to be diffused for processing. Several IoT applications are deployed using cloud computing architecture. However, cloud computing architecture is unable to meet the stringent requirements of delay-sensitive applications. Cloud paradigm being cloud-centric architecture offers high network load and latency while deploying applications on a large scale [3].

On the other hand, fog computing architecture provides resources in a distributed manner. Several benefits are there of deploying applications on fog paradigm including low latency and reduced network consumption and execution cost. Figure 1 depicts the general fog computing architecture which is a three-layer model consisting of sensor layer, fog layer, and cloud layer. The sensor layer consists of edge devices responsible for detecting the change in the environment [4]. The fog layer consists of fog devices having limited power, storage, and computational capabilities [5]. The major contribution of fog architecture is the provision of facilities near the edge devices to reduce network load and delay [6]. Deployment of IoT applications on a large scale is one of the key advantages of fog computing architecture. Fog architecture is extensively used in designing and implementing IoT applications. Several researchers have deployed the fog computing paradigm for the deployment of IoT applications. In [7], the authors have presented an energy-efficient multitier fog computing approach that introduces additional layers in the conventional fog architecture for the provision of IoT-based smart services.

The data sensed by the sensor nodes is communicated to the fog devices for processing by using the fog resources. The sensing frequency of deployed sensors determines the volume of sensed information by the edge nodes. Sufficient processing resources are required to handle the massive amount of information generated by edge devices having sensors with high sensing rates. To accomplish the processing of such a large amount of data often consumes a large part of the resources available at the fog nodes. An essential requirement to implement efficient fog-based applications is the optimized allocation of end devices to fog nodes in such a way that maintains a balance between the sensed load and available processing resources at the fog devices. In this context, an approach is proposed for fog computing-based applications to assign appropriate edge devices to parent fog nodes in such a way as to achieve a reduction in delay and network utilization.

This paper presents the design of a resource-aware load allocation algorithm for the fog-cloud computing paradigm. The proposed approach manages a balance between resources available at the fog paradigm and the volume of sensed load approaching from the sensor layer to achieve effective utilization of network resources. The proposed approach manages the processing load on fog infrastructure by assigning appropriate edge nodes to the fog devices. To

maintain optimum interconnection between the fog and edge nodes, the proposed algorithm exploits the sensing frequency information. A comparison is performed between the proposed approach and the traditional cloud-based paradigm by deploying IoT applications on multiple scales to evaluate the efficiency of the designed scheme. The performance metrics under observation during all these evaluations are network consumption and latency.

The article is divided into different sections. The segment below offers a comprehensive review of different research. Section 3 presents the system proposed. Section 4 describes the values of parameters used in the simulations. Section 5 summarizes the results and comparisons, and the last section discusses the conclusion and future directions of research.

## 2. Related Work

In this section, numerous cloud and fog computing-based systems designed related to this research work are presented.

The advent of cloud technology provides processing of a large volume of sensed data coming from geographically distributed sensors at a common point facilitating shared access to sensed information from different localities. A generic resource model keeping in view of computational resources and sensing resources is presented in [8] which facilitates request, allocation, and reservation services in sensor cloud environments.

Cloud of sensors (CoS) is a three-layer architecture that provides cloud resources near the edge for wireless sensor networks. To efficiently achieve the application requirements, several virtual nodes are assigned to applications by decoupling them from the CoS paradigm; this technique is called CoS virtualization. To resolve resource provision issues in CoS networks, the authors in [9] proposed a partially distributed algorithm, namely, Zeus, with two key functionalities. Initially, it classifies the common tasks from multiple applications and executes them once for all the applications thus minimizing the consumption of resources. Secondly, the virtual nodes are provided at the edge of the network providing reduced latency and scalability.

Three-layer Cloud of Things (CoT) is an architectural concept defining the connectivity of the IoT devices to the cloud server through edge devices. In heterogeneous and complex CoT environments, the main problem is the allocation of resources in an effective way to meet the requirements of applications. The main goal of resource allocation tasks in such environments is to maximize the number of applications to be run on the architecture using available resources. The authors in [10] presented a strategy to manage heterogeneous fog nodes scattered throughout the network to achieve efficient utilization of network resources in a way to achieve minimum delay for diverse nature of applications.

Fog architecture is providing prominent support in the implementation of delay-sensitive applications, whereas cloud computing paradigm being more resourceful was verified to be a better option in the implementation of IoT applications from a processing perspective. The authors in [11] presented a multilevel deadline-based resource

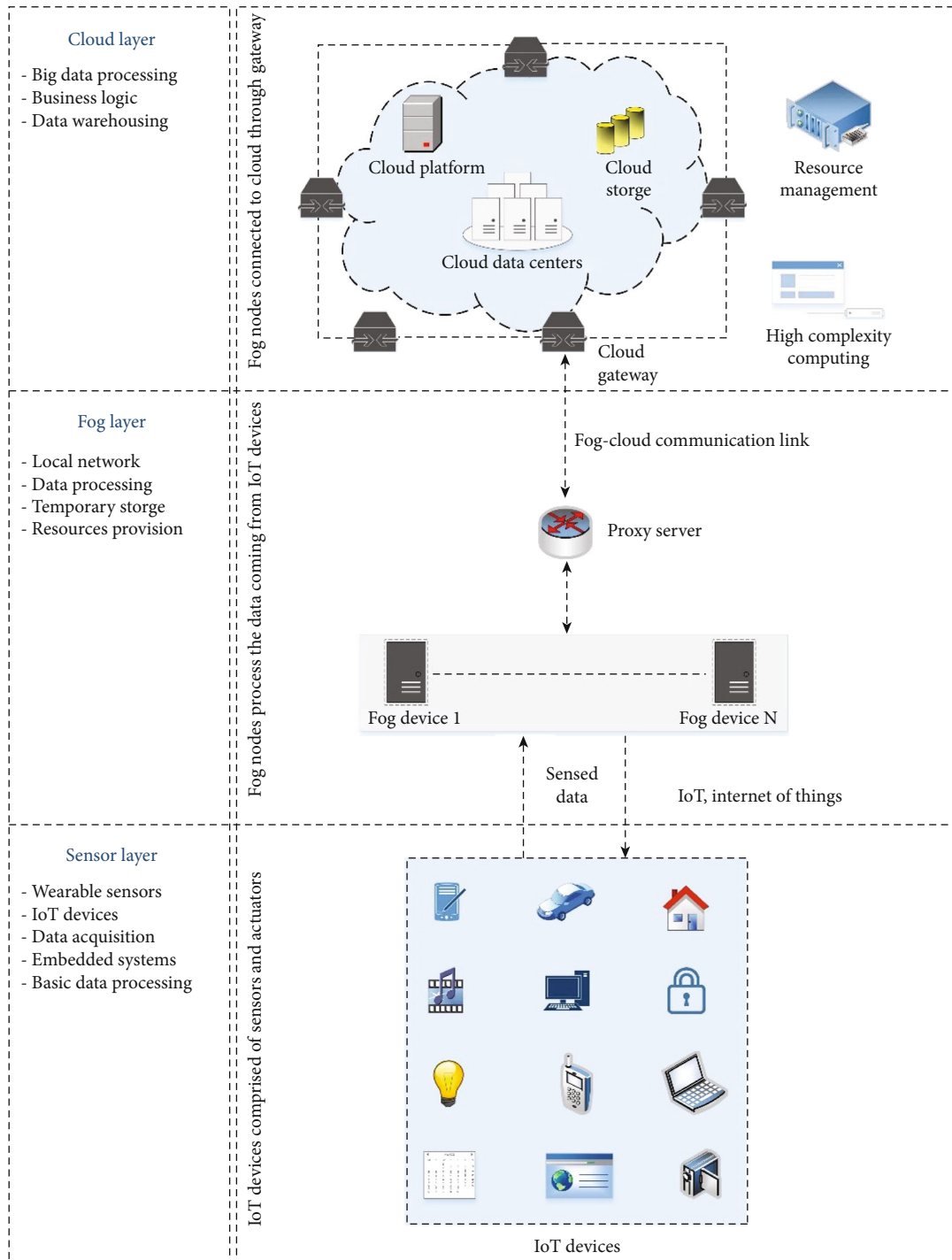


FIGURE 1: Fog-cloud computing paradigm.

allocation policy to meet dynamic user requirements for fog-cloud networks and simulated the proposed algorithm extending the CloudSim toolkit. A reduction of 12% in processing time and 15% in execution cost is achieved by using the proposed approach. In [12], the authors proposed a resource allocation model for vehicular cloud computing (VCC) networks. They model a multiobjective optimization problem with maximization of acceptance rate and minimization of the cloud provider cost as constraints. The authors

improved the nondominated sorting genetic algorithm II (NSGA-II) by amending the initial population according to the matching factor, dynamic crossover probability, and mutation probability to promote excellent individuals and increase population diversity. The results of the evaluations performed expose that the proposed methods achieve improved performance compared to the former models.

Smart cities based on several cyber-physical systems (CPS) have a different level of intelligence [13, 14]. Extensive

usage and dependency of applications on CPS can cause a disturbance, damage, and loss on failure of CPS. Cloud on the other hand can provide more sustainable solutions for smart cities. Instantaneous reception of multiple application services on sensor nodes can cause service collisions causing coupling resource management problems causing a denial of services. The authors in [15] proposed an extension of Hungarian algorithm-based fog computing architecture in which the fog layer role is to handle resource provision matters to reduce latency. The result of different experiments performed confirm that the extended algorithm provides a reduction in coupling issues and achieves efficient utilization of available fog resources.

Due to the introduction of the fog computing concept, there is a drastic increase in the deployment of latency-sensitive and large-scale applications on this paradigm. E-healthcare industry is also moving towards providing intelligent medical services to the users near to improve quality of life. Recent work in the provision of medical services using cloud and fog computing paradigm includes remote pain monitoring using cloud paradigm for newly born babies to decrease death ratio in infants [16]. In [17], the authors designed a fall detection application for aged persons using the mobile cloud paradigm and discussed the beneficial aspects of implementation healthcare applications on the said architecture. Healthcare applications have stringent QoS requirements which require low latency and stable connectivity between biopotential sensors and cloud servers. To meet latency-sensitive requirements of such applications, fog computing offers a promising solution by providing resources adjacent to the network border. In [18], the authors have deployed Medical Cyber-Physical Systems (MCPSs) on fog computing paradigm and examined different parameters affecting the cost-effective implementation of fog computing-based MCPS. They also proposed a heuristic algorithm for optimal and cost-effective implementation of fog-based MCPS.

The authors in [19] presented an energy-efficient cloud-based application for the continuous monitoring of patients in a persistent vegetative state by detecting their facial muscles. For remote monitoring access, the proposed design contains a mobile platform. The fog computing paradigm is employed for the design of a remote pain monitoring application in [20]. The presented design detects and processes the biopotential signals of patients admitted in hospitals by using fog resources and provides access to information related to patients through a web platform. The authors compared the proposed design with the cloud-based design and performed several simulations to confirm the effectiveness of the proposed design. The authors in [21] presented a fog computing-based approach for the design of an efficient car parking system that provides a reduction in latency and network consumption as compared to a cloud-based approach. A task assignment approach that allocates modules according to processing resources available at the network devices present in the system is presented in [22]. A module allocation approach for heterogeneous fog-cloud computing environments is presented in [23]. The presented algorithm efficiently allocates the application modules to the fog nodes while con-

TABLE 1: Qualitative comparison of the proposed system with the existing systems.

Reference	Paradigm	Delay	Processing cost	Network load
[19]	Cloud	High	High	High
[20]	Fog	Moderate	Low	Low
[21]	Fog	Moderate	Low	Low
[22]	Fog	Moderate	Moderate	Moderate
[23]	Fog-cloud	Moderate	Low	Low
[24]	Cloud-fog	High	Medium	Low
*Proposed	Fog-cloud	Minimum	Low	Low

TABLE 2: Comparison of the proposed algorithm with the existing resource assignment schemes.

Reference	Observed parameters during resource assignment		
	Sensing rate	Capacity	Heterogeneous devices
[22]	✗	✓	✗
[25]	✗	✓	✗
[26]	✗	✓	✗
[27]	✗	✓	✗
*Proposed	✓	✓	✓

sidering connection latency, computational power, and volume of sensed information. A comparison of the proposed approach with the traditional computing architectures is also presented by the authors. A cloud-fog based approach for the provision of on-demand health monitoring services is presented in [24]. In the presented approach, the cloud server acts as a backbone for the provision of resources and reservations for the provision of services near the edge which is resolved using the resource constraint fog nodes. Table 1 presents a qualitative comparison of the proposed approach with the existing schemes.

In [25], the authors proposed a strategy that balances the load generated by the cameras in a face recognition IoT application for video surveillance. Network usage, computational complexity, and accuracy are the performance metrics chosen for the evaluation of the proposed strategy. In [26], a QoS-aware load assignment policy is proposed that assigns latency-sensitive services on devices situated at the verge of the system to decrease cost and delay in the fog-cloud network. Primarily, the idea of a fog colony is presented in [27]. This concept is repeated in various related works. The fog colony is based on several fog cells and behaves as a micro data center. A fog cell is an application module placed on a node to offer services to linked edge nodes. In [22], the authors proposed a strategy that efficiently uses network resources and places modules according to the available capacity of network nodes. This algorithm searches for eligible network nodes to meet the module placement requirements and places modules on eligible fog nodes unless the fog layer is exhausted. Table 2 shows the comparison between the proposed algorithm and previous load assignment strategies in terms of observed parameters. The sensing frequency parameter for the approximation of

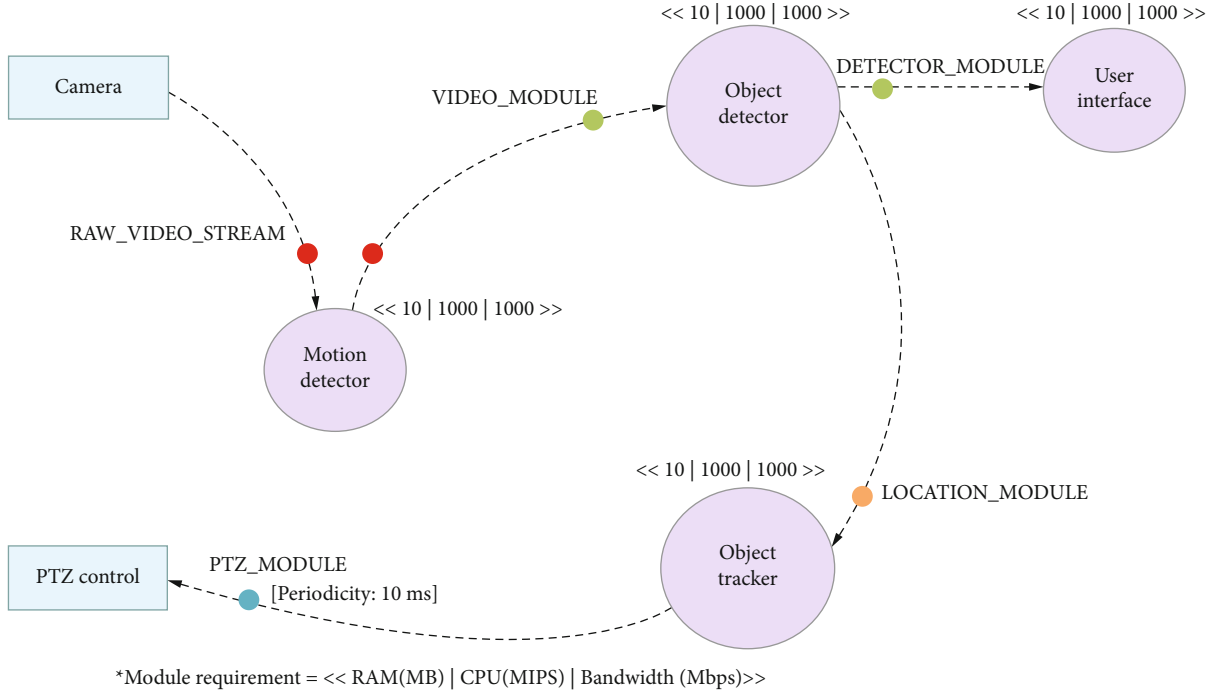


FIGURE 2: Directed acyclic graph of intelligent surveillance application.

information size is not incorporated in most of the existing models. Our proposed model also considers the heterogeneity of edge and fog devices.

### 3. System Model and Problem Formulation

Fog-cloud is an emerging concept of connecting resource deficient edge devices to resourceful cloud servers through resource-limited fog devices for implementing IoT applications. The fog-cloud architecture provides resourceful cloud servers in a centralized manner and delivers limited processing capabilities through fog devices adjacent to the sensor nodes. The fog-cloud architecture due to its distribution of resources in a decentralized manner is effective in implementing applications on large scale. This paradigm offers mobility, reduced network load, and minimum latency for implementing IoT applications. Figure 1 shows a general diagram of the fog-cloud computing paradigm in which resource-limited fog nodes are providing resources near to the edge devices. The cloud server is behaving as a centralized unit for the collection of all the information arriving from the edge nodes after preliminary processing through fog nodes. The first tier of this network is based on IoT devices which consist of sensors and actuators. The second tier consists of resource-constrained fog devices to provide basic computational functionalities and connections towards the cloud. The third tier is based on cloud servers having huge storage and computational resources available in a centralized manner. Each layer is responsible for the execution of a specific module of the application.

There is a hierarchical decrease in the resources available on each layer. The cloud server is available on top of the architecture with plenty of resources to fulfil the computa-

tional and storage demands of the applications deployed. Resource constraint fog devices are part of the second layer to link IoT devices of the first layer with the cloud on-demand. Every node in the network is responsible for the execution of some application tasks. In dynamic fog-cloud networks, the computational burden is on the fog layer as they have to provide basic processing capabilities with their limited resources to reduce latency by providing minimized load towards the cloud server.

Dynamic fog-cloud environments consist of devices with different available processing capabilities. The parameters that limit the processing functionality of any fog device are CPU and RAM. The function defining the fog node is constrained by these two parameters. If the  $i_{th}$  fog node in the paradigm is represented as  $f_i$ , then the processing capability of that node is expressed as

$$P_c(f_i) = \langle \text{CPU}_i, \text{RAM}_i \rangle. \quad (1)$$

The computational resources available in the fog paradigm is the sum of individual processing capabilities available at each fog device existing in the paradigm and is expressed as

$$N = \sum_{i=1}^M P_c\{f_i\}, \quad (2)$$

where  $M$  is the total number of fog nodes present in the network. The information sensed by the sensor nodes is transmitted to the connected parent fog device. The resources available at fog devices are consumed for the execution of tasks related to data sensed by the edge nodes. The edge

```

Input: Set of Fog devices  $F$  and Edge devices  $K$ 
Output: Assignment of appropriate edge nodes to fog devices
1: Fog devices  $f_i \in Layer\ 2$ , Edge devices  $I_i \in Layer\ 3$ 
2: for each  $I_i$  do
3:   if ( $R_i < R_{limit}$ )
4:     add  $I_i$  to  $K_L = \{\}$ 
5:   end
6:   else
7:     add  $I_i$  to  $K_H = \{\}$ 
8:   end
9: end
10: for each  $f_i$  do
11:   for  $I_i \in K_H$  do
12:     if ( $P_c(f_i) < S(I_i)$ )
13:       add  $I_i$  to  $E_i = \{\}$ 
14:        $P_c(f_i) = P_c(f_i) - S(I_i)$ ;
15:     end
16:     else
17:       for  $I_i \in K_L$  do
18:         if ( $P_c(f_i) < S(I_i)$ )
19:           add  $I_i$  to  $E_i = \{\}$ 
20:            $P_c(f_i) = P_c(f_i) - S(I_i)$ ;
21:         end
22:       end for
23:     end
24:   end for
25: end for

```

$\triangleright R_{limit}$  is predefined sensing rate limit

ALGORITHM 1: Resource-aware load allocation algorithm for a fog-cloud paradigm.

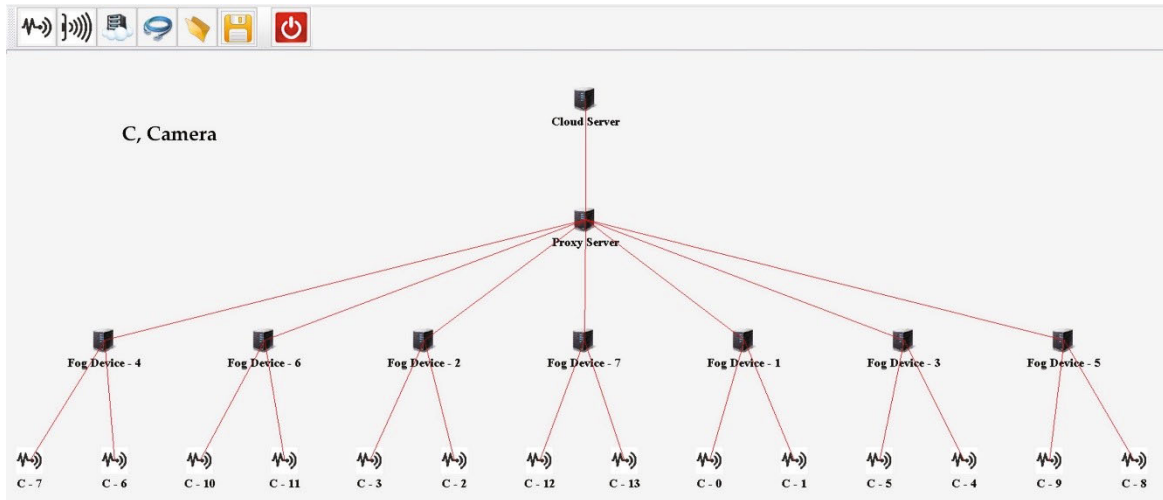


FIGURE 3: One of the scenarios created for the evaluation of the proposed paradigm.

devices are connected to the parent fog nodes to which they have to deliver the sensed information for instant processing. If there are a  $K$  number of total edge devices present in a system ( $K = \{I_1, I_2, I_3, I_4 \dots I_k\}$ ), then the edge devices connected to a fog device  $f_i$  are denoted by  $E_i$ . The sensing rate of the  $i$ <sup>th</sup> edge device  $I_{th}$  is symbolized as  $R_{th}$ . The volume of sensed information by the sensor device depends upon the sensing rate of the sensor attached to the end devices. In the fog-cloud paradigm, the received data demanding high processing resources than the available at

fog nodes is communicated to the parent cloud for execution. To achieve optimum performance of the fog-cloud paradigm, the volume of sensed load at fog nodes is an important parameter to be considered. This parameter was not exploited in literature work published before related to resource utilization and allocation in fog-cloud paradigms. For the  $i$ <sup>th</sup> edge device,  $I_i$ , the total volume of sensed data is directly proportional to the sensing rate of the sensor attached and is expressed as  $S(I_i)$ . Consequently, the volume of sensed data that a fog device has to process depends on

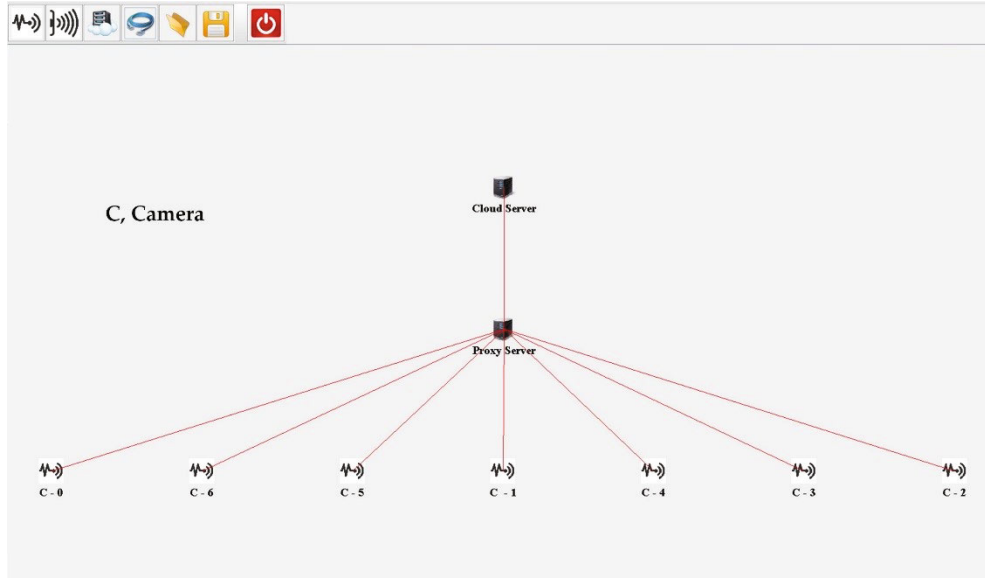


FIGURE 4: One of the scenarios created for the evaluation of the cloud-based approach.

TABLE 3: Network configurations related to network nodes used in simulations.

Specification	Cloud server	Gateway node	Fog device	End device
Rate per execution	0.01	0.0	0.0	0.0
Random access memory (MB)	40000	4000	2000 - 4000	1000
Downlink capacity (MB)	9.7656	9.7656	9.7656	9.7656
Processing power (MIPS)	448100	2800	2000 - 4000	500
Uplink capacity (MB)	0.0	9.7656	9.7656	9.7656

the sensed information volume at each edge device connected to that fog device which is given by

$$L(f_i) = \sum_{\forall I_i \in E_i} S(I_i). \quad (3)$$

For optimum performance of fog-cloud computing paradigm, the connection of appropriate edge devices according to available resources at parent fog device plays a pivotal role.

In the traditional deployment of applications on the fog-cloud computing paradigm, the edge devices are connected to fog devices irrespective of the sensing rate of sensors at the edge nodes which results in high network consumption and latency. In such situations, fog nodes having excess processing resources to handle the high volume of information might have a connection to edge nodes producing a low volume of data resulting in wastage of fog resources. Consequently, the resource constraint fog nodes having a connection to edge nodes with high volumetric production of sensed information can overburden the fog nodes to tackle such information. In this research, a policy is proposed that deploys edge nodes according to the availability of resources at the fog layer to reduce network burden and delay for efficient utilization of the fog-cloud network resources.

TABLE 4: Types and characteristics of tuples used in simulations.

Tuple	Tuple length	CPU length (MIPS)
PTZ_MODULE	28	100
VIDEO_MODULE	2000	2000
LOCATION_MODULE	100	1000
DETECTOR_MODULE	2000	500
CAMERA	20000	1000

This research redefines and simulates the intelligent surveillance application using distributed camera networks [28] to demonstrate the effectiveness of the proposed scheme. For the better understanding and representation of application components in a distributed computing environment, Distributed Data Flow Model (DDF) [29] was used in this research for the deployment of applications on the fog-cloud paradigm. Figure 2 presents the directed acyclic graph (DAG) of the intelligent surveillance application based on a distributed network of cameras deployed for the evaluation of the proposed algorithm. In the DAG model, the five modules of the deployed application are represented as a vertex that processes the information approaching from the preceding module. The arcs linking various modules describe the flow of data between different modules.

The DAG model of the application consists of five modules as shown in the above diagram. The motion detection

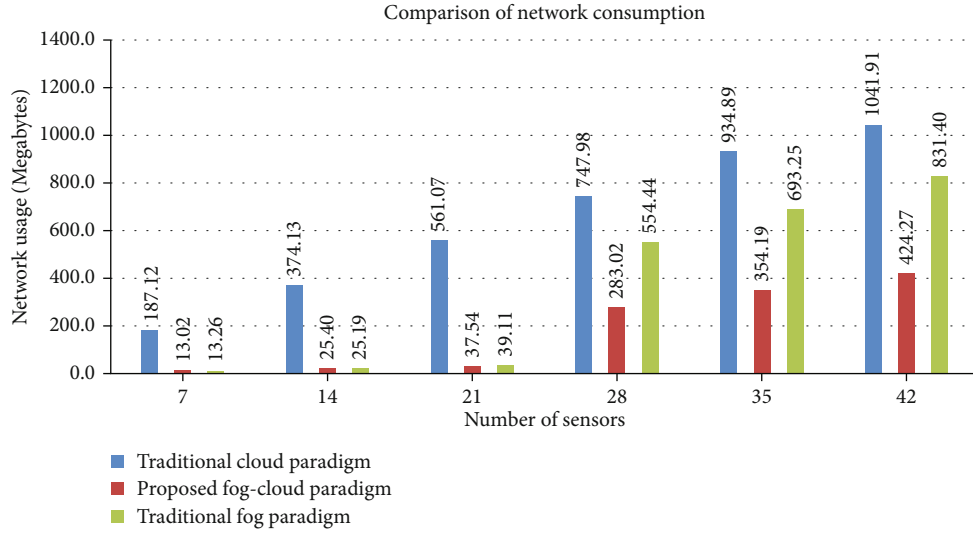


FIGURE 5: Comparison of network consumption.

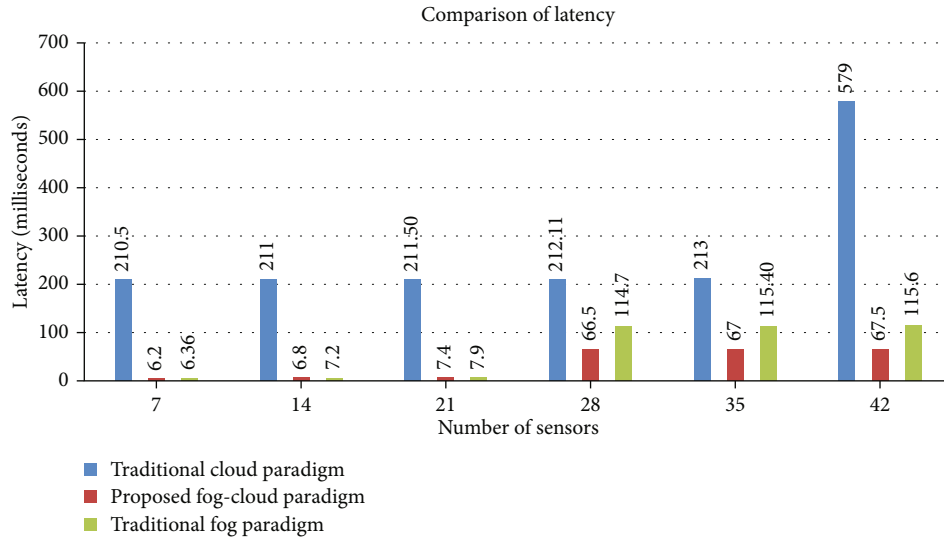


FIGURE 6: Comparison of latency.

module is placed in cameras for capturing video streams for detecting the motion of an object. This module forwards the information to the object detector module on the detection of motion of an object. Afterwards, the object detection module tracks the object and calculates the coordinates of the object. The object tracker module receives the coordinates previously calculated by the object detector modules and calculates the PTZ configurations for the camera for effective monitoring of the detected object. The PTZ control module after receiving the PTZ configurations adjusts the cameras accordingly. Finally, filtered video streams collected from the object detector module are conveyed to the user's device via the user interface module for better visualization of the tracked object. In the computing atmosphere, the basic unit used for intramodule communication is a tuple that is characterized by its specific length containing information to be processed and the number of resources required for its processing. Tuple mapping is described in

DAG using coloured circles. For example, reception of a tuple of type RAW\_VIDEO\_STREAM on the module "Motion Detector" will result in a release of tuple type VIDEO\_MODULE.

#### 4. Proposed Solution

In this paper, a resource-aware load allocation algorithm is proposed that effectively manages the connection between edge nodes and parent devices by taking into account the available processing resources at the fog layer and volume of sensed information at end devices. The proposed algorithm allocates appropriate edge devices to each fog device present in the network after searching throughout the edge layer. To balance the processing load on fog nodes according to their computational resources, the algorithm effectively places edge nodes under fog nodes according to the sensing rate of sensors present at the edge nodes. The proposed approach classifies



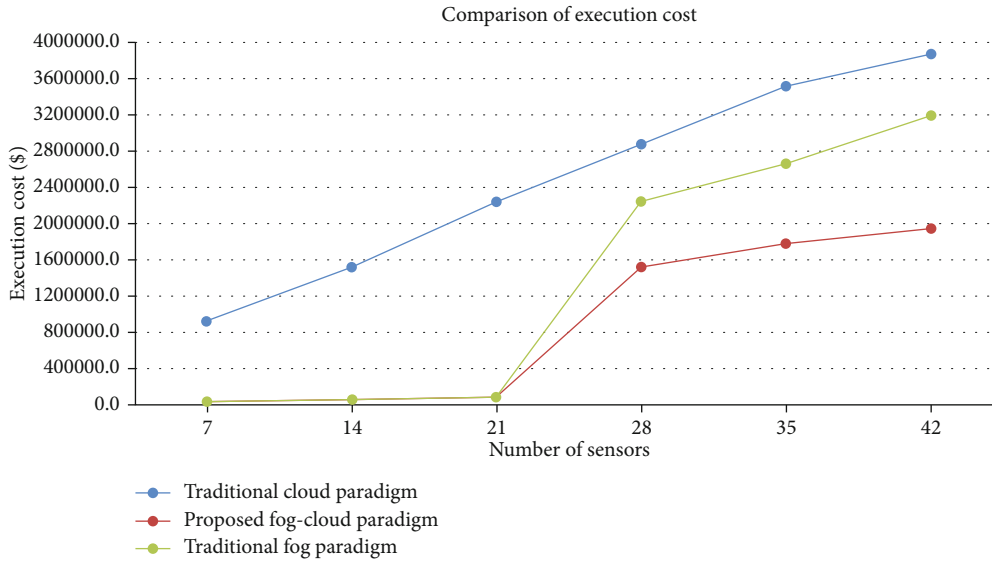


FIGURE 7: Comparison of execution cost.

and registers the edge devices as edge nodes with high sensing rates and low sensing rates. Afterwards, from the categorized edge nodes, a combination of edge nodes is allocated to the fog devices according to resources existing at the fog nodes. The proposed Algorithm 1 is given below.

Fog nodes and edge devices are given as input to the proposed algorithm. Initially, the algorithm categorizes the edge devices on the basis of sensing rates of the sensors attached to these edge nodes. If the sensing rate of the edge device is less than the predefined rate, then it is placed in the set  $K_L$ , otherwise the edge device is placed in the set  $K_H$ . Afterwards, the algorithm searches throughout the set  $K_H$  and  $K_L$  for the selection of appropriate edge devices. The algorithm assigns suitable edge devices to fog nodes for optimal performance. If the resources required for the processing of the sensed volume by an edge device are less than the available at the fog device, then that edge device is assigned to that fog device as child node.

## 5. Results and Discussion

For validating the effectiveness of the proposed scheme, an intelligent surveillance application is implemented on different scales. In each experimental scenario, the number of cameras monitoring the area under surveillance is increased. There are a total number of seven areas that are under surveillance during all the simulations. In all the simulated scenarios, the cameras are connected to fog nodes that are associated with the cloud server. One fog device per area under surveillance is assigned which provides resources close to the boundary of the network to observe and detect activity in that area. The fog nodes provide resources near to cameras for the processing of video streams recorded by cameras. The number of cameras per surveilled area is varied in each physical topology created in the simulation. Initially, two cameras per fog node are connected which are increased in each new topology created. The sensors created in the simulations are according to the strategy of [30]. In each sce-

nario, the number of cameras per fog node is increased to analyze the delay and network consumption. The view of one of each scenario created in iFogSim for the proposed paradigm and traditional cloud computing paradigm is shown in Figures 3 and 4.

The tabulated network configurations, tuple types, simulation parameters, and sensing frequencies used in the simulations are shown in Tables 3 and 4. For evaluation, the intelligent surveillance application is implemented on the proposed load aware resource allocation fog-cloud paradigm and the traditional cloud and fog computing paradigms. The information detection frequencies of cameras deployed in the simulations take values between 5 ms and 20 ms.

A comparative analysis is performed between the proposed scheme and traditional cloud and fog paradigms by creating various simulation scenarios on multiple scales. Cost of processing at cloud, network consumption, and end-to-end delay are the parameters under observation during all these evaluations. Figure 5 presents a comparison between the consumption of networks during implementing the application on different paradigms. The proposed algorithm successfully decreases the load on the network as compared to fog and cloud-based implementations.

In the proposed approach, the parent fog nodes are assigned to the child devices according to the volume of data sensed by them. The volume of the sensed load is associated with the sensing rate of the sensing device, so edge devices having higher sensing rates are assigned to the fog nodes having higher data processing capacity to reduce the load on the network. Therefore, the load is aligned with the network resources available which in turn decreases the overall burden on the system. On the contrary, in the cloud architecture, all the sensed load is handed directly to the cloud server for processing resulting in high network utilization. Due to the inability of provision of fog resources according to sensed load, the traditional fog paradigm provides reduced network consumption as compared to the cloud paradigm but is not much network efficient as compared to the proposed model.

A comparison between the proposed paradigm with the traditional cloud and fog paradigms in terms of offered latency is presented in Figure 6. In cloud-based implementation, all the sensed data from the system is to be processed by the cloud server; thus, rise in latency is proportional to the number of sensors linked to the cloud, whereas in the fog computing paradigm, the processing of data is also provided at the intermediate level by the fog nodes resulting in a decrease in the information to be processed by the cloud server and decreasing the round-trip time. The core feature of the designed strategy is to reduce the latency and burden on the network by allocating the suitable fog resources to the edge nodes according to the rate at which the information is sensed by the edge nodes. The proposed model estimates the volume of information arriving from the edge devices by analyzing the sensing frequency of the sensors installed at the edge devices. Afterwards, the algorithm links the suitable fog devices to the sensor nodes according to the resources of the fog devices. The proposed policy reduces the amount of data to be processed at the cloud server by providing suitable fog resources according to the demand from the edge devices which in turn decreases the processing cost at the cloud as depicted in Figure 7.

The proposed algorithm estimates the volume of information generated by the edge nodes by incorporating the sensing rate information of the sensors attached to these nodes. Afterwards, the suitable edge nodes are linked to the parent fog devices. This provision of fog computing resources according to the volume of sensed information is not offered in traditional fog computing designs. This optimum balance between the detected volume of information to be processed at edge nodes and the processing capacity available at the parent fog device is the key feature of the proposed algorithm. Due to this salient feature, most of the information generated at the sensor nodes is processed at the linked fog device resulting in a reduction of information to be processed at the cloud server. Thus, a significant reduction in the cost of execution is observed as compared to traditional fog designs.

## 6. Conclusions

The algorithm proposed in this research effectively balances the available resources offered by the fog paradigm and the volume of information generated at the edge of the network. To achieve this, the proposed scheme manages the connection between the fog nodes and edge devices. The proposed strategy estimates the volume of information detected at the edge nodes and assigns appropriate parent fog devices from the available nodes at the fog paradigm. This efficient management of sensed load and processing resources of the network by the proposed algorithm effectively reduces the latency and network consumption of the system. The intelligent surveillance through distributed camera network application was implemented on different scales to compare the proposed algorithm with the traditional cloud and fog architectures. iFogSim toolkit is used to perform these simulations. The results of the comparison show that the proposed algorithm prominently reduces the processing cost

at cloud, delay, and network consumption. The proposed strategy is capable to execute any type of application. My future work consists of deploying more applications on the proposed design and modification of the proposed algorithm for the examination of multiple parameters. Moreover, future research includes the analysis and design of expected glitches triggered due to node failure in the system.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Authors' Contributions

Syed Rizwan Hassan and Ishtiaq Ahmad contributed to actualization, validation, methodology, formal analysis, investigation, software, and initial draft. Ateeq Ur Rehman, Seada Hussen and Habib Hamam contributed to actualization, validation, methodology, formal analysis, investigation, and initial draft. All authors read and approved the final version.

## References

- [1] A. S. Petrenko, S. A. Petrenko, K. A. Makoveichuk, and P. V. Chetyrbok, "The IIoT/IoT device control model based on narrow-band IoT (NB-IoT)," in *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp. 950–953, Moscow and St. Petersburg, Russia, 2018.
- [2] H. Shahid, M. A. Shah, A. Almogren et al., "Machine learning-based mist computing enabled Internet of Battlefield Things," *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 4, pp. 1–26, 2021.
- [3] M. Villari, M. Fazio, S. Dustdar, O. Rana, and R. Ranjan, "Osmotic computing: a new paradigm for edge/cloud integration," *IEEE Cloud Computing*, vol. 3, no. 6, pp. 76–83, 2016.
- [4] K. A. Awan, I. U. Din, A. Almogren, H. A. Khattak, and J. J. Rodrigues, "Edge trust-a lightweight data-centric trust management approach for green internet of edge things," *Wireless Personal Communications*, 2021.
- [5] V. Moysiadis, P. Sarigiannidis, and I. Moscholios, "Towards distributed data management in fog computing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 7597686, 14 pages, 2018.
- [6] F. Firouzi, B. Farahani, and A. Marinšek, "The convergence and interplay of edge, fog, and cloud in the AI-driven Internet of Things (IoT)," *Information Systems*, vol. 107, article 101840, 2021.
- [7] M. Ammad, M. A. Shah, S. U. Islam et al., "A novel fog-based multi-level energy-efficient framework for IoT-enabled smart environments," *IEEE Access*, vol. 8, pp. 150010–150026, 2020.
- [8] S. Bose, D. Sarkar, and N. Mukherjee, "A framework for heterogeneous resource allocation in sensor-cloud environment," *Wireless Personal Communications*, vol. 108, no. 1, pp. 19–36, 2019.

- [9] I. L. Santos, L. Pirmez, F. C. Delicato et al., “Zeus: a resource allocation algorithm for the cloud of sensors,” *Future Generation Computer Systems*, vol. 92, pp. 564–581, 2019.
- [10] T. C. Xavier, I. L. Santos, F. C. Delicato et al., “Collaborative resource allocation for Cloud of Things systems,” *Journal of Network and Computer Applications*, vol. 159, article 102592, 2020.
- [11] R. K. Naha, S. Garg, A. Chan, and S. K. Battula, “Deadline-based dynamic resource allocation and provisioning algorithms in Fog- Cloud environment,” *Future Generation Computer Systems*, vol. 104, pp. 131–141, 2020.
- [12] W. Wei, R. Yang, H. Gu, W. Zhao, C. Chen, and S. Wan, “Multi-objective optimization for resource allocation in vehicular cloud computing networks,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2021.
- [13] K. Haseeb, I. U. Din, A. Almogren, I. Ahmed, and M. Guizani, “Intelligent and secure edge-enabled computing model for sustainable cities using green internet of things,” *Sustainable Cities and Society*, vol. 68, article 102779, 2021.
- [14] I. U. Din, A. Bano, K. A. Awan, A. Almogren, A. Altameem, and M. Guizani, “LightTrust: lightweight trust management for edge devices in industrial Internet of Things,” *IEEE Internet of Things Journal*, 2021.
- [15] T. Wang, Y. Liang, W. Jia, M. Arif, A. Liu, and M. Xie, “Coupling resource management based on fog computing in smart city systems,” *Journal of Network and Computer Applications*, vol. 135, pp. 11–19, 2019.
- [16] S. Tejaswini, N. Sriraam, and G. Pradeep, “Cloud-based framework for pain scale assessment in NICU—a primitive study with infant cries,” in *2018 3rd International Conference on Circuits, Control, Communication and Computing (I4C)*, pp. 1–4, Bangalore, India, 2018.
- [17] F. Muheidat, L. Tawalbeh, and H. Tyrer, “Context-aware, accurate, and real time fall detection system for elderly people,” in *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, pp. 329–333, Laguna Hills, CA, USA, 2018.
- [18] H. K. Apat, K. Bhaisare, B. Sahoo, and P. Maiti, “Energy efficient resource management in fog computing supported medical cyber-physical system,” in *2020 International conference on computer science, Engineering and Applications (ICCSEA)*, pp. 1–6, Gunupur, India, 2020.
- [19] B. Gj, “Internet of Things (IoT) and cloud computing based persistent vegetative state patient monitoring system: a remote assessment and management,” in *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, pp. 301–305, Belgaum, India, 2018.
- [20] S. Shukla, M. F. Hassan, M. K. Khan, L. T. Jung, and A. Awang, “An analytical model to minimize the latency in healthcare internet-of-things in fog computing environment,” *PLoS One*, vol. 14, no. 11, article e0224934, 2019.
- [21] K. S. Awaisi, A. Abbas, M. Zareei et al., “Towards a fog enabled efficient car parking architecture,” *IEEE Access*, vol. 7, pp. 159100–159111, 2019.
- [22] M. Taneja and A. Davy, “Resource aware placement of IoT application modules in Fog-Cloud computing paradigm,” in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 1222–1228, Lisbon, Portugal, 2017.
- [23] S. R. Hassan, I. Ahmad, J. Nebhen, A. U. Rehman, M. Shafiq, and J.-G. Choi, “Design of latency-aware IoT modules in heterogeneous fog-cloud computing networks,” *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 70, no. 3, pp. 6057–6072, 2022.
- [24] C. S. Nandyala and H.-K. Kim, “From cloud to fog and IoT-based real-time U-healthcare monitoring for smart homes and hospitals,” *International Journal of Smart Home*, vol. 10, no. 2, pp. 187–196, 2016.
- [25] S. S. N. Perala, I. Galanis, and I. Anagnostopoulos, “Fog computing and efficient resource management in the era of Internet-of-Video Things (IoVT),” in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, Florence, Italy, 2018.
- [26] S. Azizi, F. Khosroabadi, and M. Shojafar, “A priority-based service placement policy for fog-cloud computing systems,” *Computational Methods for Differential Equations*, vol. 7, pp. 521–534, 2019.
- [27] O. Skarlat, M. Nardelli, S. Schulte, M. Borkowski, and P. Leitner, “Optimized IoT service placement in the fog,” *Service Oriented Computing and Applications*, vol. 11, no. 4, pp. 427–443, 2017.
- [28] B. Song, C. Ding, A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, “Distributed camera networks,” *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 20–31, 2011.
- [29] M. Ashouri, P. Davidsson, and R. Spalazzese, “Quality attributes in edge computing for the Internet of Things: a systematic mapping study,” *Internet of Things*, vol. 13, article 100346, 2021.
- [30] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and 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, pp. 1275–1296, 2017.