National College of Ireland

Computation Based Load Balancing Algorithm For improving Quality of Service

MSc Research Project
Cloud Computing

Satya Venkata Dinesh Kumar Vetcha
Student ID: 20238304

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: …Satya Venkata Dinesh Kumar Vetcha ……

Student ID: ……20238304…….……

Programme:……Cloud Computing …… **Year:** ……2022……..

Module: ……MSc Research Project ……..……

Supervisor: ……Sean Heeney……..……

Submission Due Date: …19/09/2022……..……

Project Title: Computation Based Load Balancing Algorithm For Improving Quality of Service……..……

Word Count: ……6069…… **Page Count**……19……..……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……

Date: …19/09/2022……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Computation Based Load Balancing Algorithm for Improving Quality of Service

Satya Venkata Dinesh Kumar Vetcha
20238304

Abstract

As technology is evolving, the usage of IoT devices is increasing gradually in our life. The requests are sent to the cloud computing that arises problems like high latency and bandwidth. So, fog environment is introduced, and it is an extended version of cloud computing. Fog computing usage is also increased, and it also needs to be standardized. The main idea of the fog environment is to provide a better quality of service by utilizing the resources by distributing the load on the fog nodes. To provide a better quality of service and also to distribute the load on fog nodes, we are proposing a computational-based load balancing algorithm. If the fog node meets the load, then it will check the capability of the fog nodes for performance then if the node meets the capacity then it will assign otherwise the tasks are assigned to the next nearest high computational fog node. For the proposed algorithm the iFogSim simulator is used for the simulation and written in a java programming language. The results and experiments are carried out by taking the 6 to 10 virtual machines into consideration and also compared with the existing algorithms. By conducting experiments, I have shown the comparison of proposed and default algorithms.

1 Introduction

With the improvement in technology, cloud computing is become to play an important role in our life. It transformed many industries and organizations as well as helped to improve the growth of the company. Cloud computing has become one of the most valuable research areas in the world of computing [1]. These are a combination of high computation power, and large storage, as well as they, can provide on-demand service, scalability, and many more. We can access the data and services from any location on the globe. These cloud data centers are placed far from the end-users. These are far from users generating an issue like latency and required more bandwidth.

As the usage of IoT devices increases the data generated by these sensors also increases and the load on the cloud also increased. In order to overcome these issues, CISCO has introduced Fog Computing. It is an extension of cloud computing. Fog computing means the creation of virtual machines, which are capable of performing operations, storage, and network and also provide other services which are placed closer to the end-users. It acts as a middle layer between the cloud and IoT devices. By using fog computing load on the cloud is decreased, increases the efficiency, and speed of communication(network), and also improves the performance of handling the IoT's data. Fog computing can provide a quick response for the end-users. This leads to a decrease in the load on the cloud and using the load balancing

algorithm that can be able to distribute the load equally on the distributed computing by providing a better quality of service(QoS).

Fog computing is placed between the cloud and the IoT. As the IoT technology is evolving the requests are sent to the fog nodes that are able to provide a quick response with low latency and use less bandwidth. By utilization of the load balancing algorithm and scheduling of tasks on the fog devices help to provide a quality of service for the end-users. The below figure represents the architecture of fog computing.

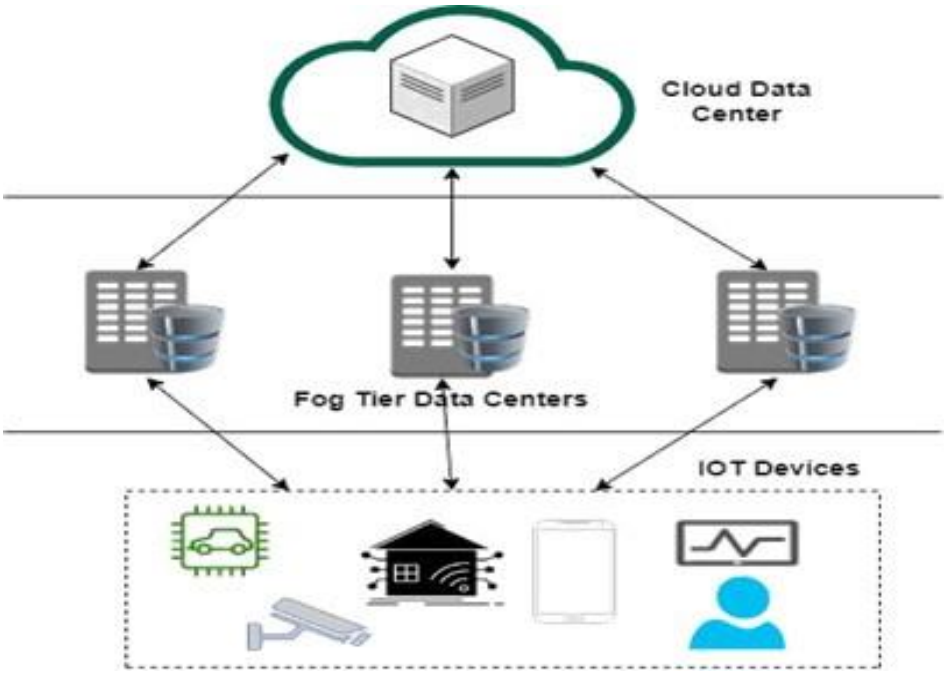

Figure 1: Fog Computing 3-Tier Architecture

1.1 Motivation

The Internet of things usage is increasing day by day and these devices play an important role in our daily life. As these devices usage is increasing and also the data generated by these devices lead to the development of cloud computing. These data centers are a combination of high computation, storage, network, and many other resources. But these are placed far from the users. These are rising issues like delay, high energy consumption, and also increasing the load on the cloud. The usage of IoT devices increasing and they are generating about 70000 requests that are sent to the cloud and which leads to the delay. Because they are placed far from users. In order to overcome these issues, fog computing came into existence, and they are placed near the end-users.

Based on the survey**,** authors[2] much research has been conducted in the fog environment. The main idea of this research is to provide better QoS services and also resource management. As the usage of IoT devices usage is increasing, they are generating a huge amount of data and requests to the fog nodes. By using the fog nodes, the number of requests that are transferred to the cloud is decreased. This helps to reduce the traffic and also decreases resource usage, improving the quality of service for the end-users [1]**.** By sending the request to the cloud it has rising issues like latency and also many other issues. Many

issues are resource allocation for assigning the task on the fog nodes and also the quality of service needs to be improved.

Much research is done and performed related to resource management and also the quality of service in a fog environment. In which different performance and surveys are addressed and they are able to show the issues related to resource management. The main idea of the research is to develop, design, and implement the computation-based load balancing algorithm for the better assigning of the tasks in distributed computing based on the resources of the server.

1.1 Research Question

Fog computing is an extended version of cloud computing. It has extended all the services like storage, network, computation, and many more. The application operating on fog computing to be distributed equally on all the fog nodes is become the main challenge and then which leads to providing improved quality of service for end-users.

"How can the load be distributed evenly throughout the fog nodes based on the computation capability if the nearest fog nodes are overloaded?"

2 Related Work

2.1 Survey of Fog Computing

In today's world, technology is developing the usage of IoT devices is also increased a lot. The data and requests generated from sensors, and actuators also improved. This leads to the development of cloud computing, fog computing, and edge computing. Many challenges like security, energy management, resource management, infrastructure design, failure management, and designing standard architecture.

By reviewing the paper, the authors [3] did the most research on IoT devices in the area of fog computing. Because the IoT devices are generating a huge volume of data, and this is able handled by cloud computing. They discussed related to the processing of data generated and also the issues in cloud computing. This survey mostly talks about the fog computing importance and also it is compared with the cloud computing characteristics. They suggested that fog computing helps to decrease the time for data transfer, latency, and bandwidth. At the end of the paper, the fog computing challenges, and the main challenge is the distribution of load on fog nodes.

Fog computing is the most widely researched area in recent years. The authors [4] did research in the fog environment to improve the quality of service(QOS) such as scalability, response time, latency, bandwidth, and security research areas. By reading this paper, we can understand the current trends and future directions related to fog computing. In this paper, mainly discussed issues related to the QoS metrics, applications, and implementation details. In the end, the challenges are improving the service level agreements and resource scheduling leads to better improvement in fog computing.

As the usage of IoT devices is increasing they are generating a huge amount of data from the sensors. To handle this cloud computing is used. The authors [5]**,** say cloud computing is able to handle data. But in some applications like healthcare need a quick response. The cloud data centers are placed far from the end-users and give delays in response. They

suggested fog computing is a suitable method and also improves the quality of services. They designed three-layer architecture for patient's heart monitoring systems. The designed framework also gives importance to resource allocation and management for the improved quality of service(QoS) in healthcare applications. By the end of this paper, fog computing is the best suitable scenario for healthcare applications.

2.2 Load Balancing Algorithms in Fog Computing

The usage of IoT devices is increasing in our life. The usage of cloud computing also increased, and the requests are also increased. These lead to the issues like delay and many more. To avoid these issues and improve the quality-of-service fog computing came into existence. The fog nodes are able to give a quick response when compared with the cloud. The fog computing also needs to be standardized and the load on this is also distributed equally then which leads to providing an improved quality of service for the end-users. In this paper, the authors [6] have done research about the load balancing algorithms and the simulation tools that are required. They divided the load balancing algorithms into different classifications, exact, including approximation, fundamental, and hybrid. The widely used simulation tools in the fog environment are cloud analyst, iFogSim, omnet++, and Matlab. By reviewing them, I can understand that the allocation of load on the fog nodes by load balancing algorithms helps to improve the quality of services like cost, resource utilization, response time, throughput, energy consumption, and performance.

Fog is becoming the most popular with the increase of IoT devices and their usage. As the usage is increased, the load and issues related to cloud computing are overcome. The usage of IoT devices is increasing in all the areas like driverless cars, healthcare, and many more. These areas require a quick response in that case fog computing is the best-suited environment for the cloud. In order to provide a quick response and also to distribute the load on the fog environment. The authors [7], designed a load balancing algorithm called fuzzy golden eagle load balancing [FGELB]. They received tasks performed in three stages, they are power management, assigning the task based on priority, and the power management. By considering different characteristics they have given priority to tasks by using the fuzzy algorithm, and also, they used the golden eagle optimization algorithm (GEOA) for scheduling resources and ranking the tasks. The GEOA algorithm helps to execute the tasks with the required resources. They conducted different experiments and compared them with the existing algorithms with the characteristics like computational time, failure rate, waiting time, energy consumption, and communication overhead. This algorithm performed well, but to execute these tasks more resources are consumed.

In this paper, the authors [8] proposed two distinct natural-inspired scheduling techniques for the fog nodes that have been suggested. The two algorithms were created with the primary goal of enhancing the quality of service by concentrating on fog node response time. They took the comparison of designed particle swarm optimization (PSO) and ant colony optimization (ACO) outputs with the default round-robin algorithm like communication time and the response time characteristics for a better QoS. The ant colony optimization algorithm takes the shortest way to perform the task and also next task is also performed in the same way. When comparing the results PSO algorithm performs better than the default round-robin algorithm but not more than the ACO. They conducted different experiments by taking smart city implementation and also load is also distributed equally on all the fog nodes. In future work, I can try to implement this algorithm by taking different scenarios and also try to

reduce the other parameters like computation time, energy consumption, and many other things.

Fog computing is an extended version of cloud computing with storage, computation capability, network, and many more. To provide a solution, the authors [9] proposed and designed an energy-aware load balancing algorithm for the allocation of tasks on the fog nodes based on the energy level of the fog servers. They are shown as the flowchart of the proposed algorithm and used iFogSim simulator for simulation. The proposed algorithm is compared with the default algorithm and in the results, the energy is consumed less by the existing algorithms. The main idea of the proposed energy-based algorithm is to utilize fewer power resources and that leads to better resource utilization and also improved quality of service. Finally, they are taken the energy level into consideration for assigning tasks.

In this paper, the authors [10] found that the dynamic resource allocation algorithm is not an energy-efficient method to overcome these issues they proposed an energy-efficient resource allocation method. The proposed method allocates the resources for the request based on the required energy and power consumption. In order to assign the task, they are few steps are performed. The resource information provides the resources based on the instruction in the task and then the task manager transfers the task to the resource scheduler which helps to filter the tasks in ascending order based on the computation power and energy consumption. The results are performed by taking multiple fog nodes into consideration and also compared with the dynamic resource allocation method (DRAM). When the results are compared with the existing DRAM the proposed method performs better. This method is not a fault tolerance method.

The major goal of the designing of load balancing algorithms is to distribute the load in distributed computing. The authors [11], designed the min-min algorithm. They implemented smart grid technology the improvement of reliability, stainability, and efficiency in the fog environment. They integrated smart grid technology in fog and cloud environments. They performed analysis by taking 6 fog nodes in 6 different regions and they are integrated with the microgrids for the calculation of energy consumption on these fog nodes. The designed algorithm completes the task which takes less execution time and the received tasks which takes more time for execution will stay in the queue. Based on the completed tasks and newly received tasks the list is updated. The designed algorithm gave importance to the minimum execution time. The tasks which require a quick response with a longer execution time have to be in a queue.

By extending cloud services like networking, storage, and computation to the end users is called fog computing. Based on these issues, the authors [12] proposed an energy optimization method in a fog environment with efficient work scheduling. The proposed artificial bee colony (ABC) is an optimization technique with a combination of artificial neural networks (ANN). The suggested method for scheduling algorithm helps to filter the virtual machines and choose the most appropriate for the tasks based on energy. The selected virtual machines are assigned for the ABC algorithm. The ANN technique is used to reduce the failure rate while performing the computation. The load on the virtual machines is not considered while constructing the algorithms, even if it is able to accomplish energy optimization, which increases end-user delays.

2.3 Overview Literature Review

Load balancing algorithms literature review overview Table 1.

Table 1: Summery of Load Balancing Algorithms

Authors	Algorithm	Main Idea
[7]	Fuzzy Golden Eagle Load Balancing (FGELB) and Golden Eagle Optimization Algorithm(GEOA)	The Tasks given priority based on the task size, deadline time. Those tasks are executed first.
[8]	Particle Swarm Optimization(PSO) and Ant-Colony Optimization (ACO) scheduling Algorithms	The short path is chosen by the ACO algorithm to complete the task and PSO algorithm uses task offloading methods.
[9]	Energy-Aware Load Balancing Algorithm	The tasks are based on the energy level of the fog nodes.
[10]	Dynamic Energy Efficient Resource Allocation(DEER) load balancing technique	The tasks will distributed according to which ones use the least amount of energy in ascending order.
[11]	Min- Min Scheduling Algorithm	The tasks that take the least amount of time to complete will be carried out first, and the list is revised each time.
[12]	Artificial Bee Colony (ABC) technique	Based on the storage the fog nodes are taken into consideration.

3 Research Methodology

The usage of IoT devices like sensors and also mobile device usage is also increased. This leads to increases in the usage of cloud computing, and it plays a huge role in the industry. cloud computing is extremely important in processing, data management, security, and storage. Cloud computing is able to handle these operations, but they are placed far from users. The distance leads to delays, power consumption, and bandwidth usage increases a lot. To overcome these issues and provide a better quality of service with less utilization of resources fog computing came into existence. As the usage of IoT technology is increasing, they are generating a huge amount of data, and also load distribution in the fog environment is also becoming an issue. Because of this, providing a better quality of service in the fog environment has also become a huge challenge.

Many developers are also trying to resolve issues like load distribution, latency, and performance. Many giant companies are providing a public cloud with pay as you go service model for many companies and also for the customers. To resolve the issue like load distribution in the fog environment there are many tools like FogBus, iFogSim, a cloud analyst, CloudSim, and much more available. For the proposed method, load distribution on virtual machines and also providing a better QoS service iFogSim simulator is the best suited for the research method. The proposed methodology is shown in below figure 2.

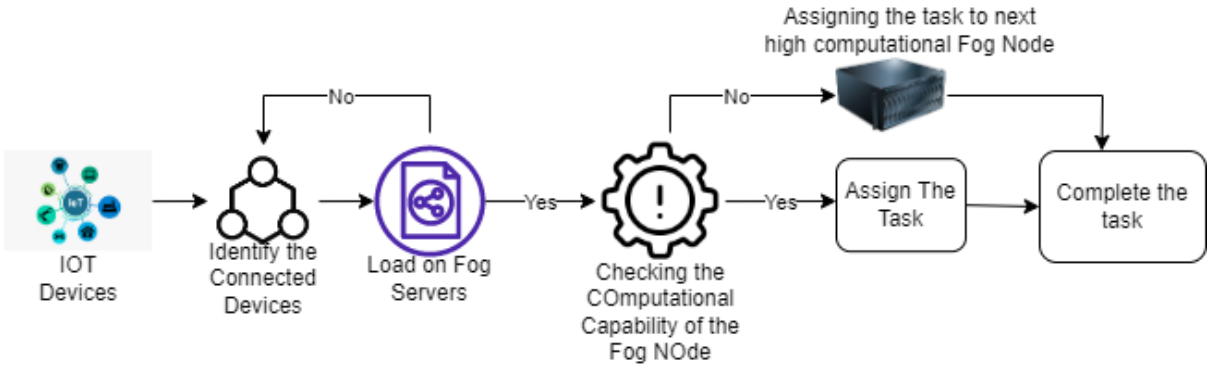

Figure 2:- Proposed Methodology Overview

3.1 iFogSim Simulator

Many researchers have done fog computing there are trying to resolve the issues like load distribution on fog nodes and also provide an improved QoS service. Implementing these proposed methods practically is very tough. To implement the proposed method, they are many simulators are existed in fog computing. The proposed method comes under resource management for this method iFogSim simulator is used for the research purpose. This is an enhanced version of the CloudSim simulator. It is having all the basic functionalities that are available from the CloudSim simulator like the creation of virtual machines with different areas, and these VMs can communicate with each other [13]. This simulator is compatible with the cloud, fog computing, and also for IoT devices. The simulator is used for resource management, and also for the SLA's means service allocation policies. The following primary key components are used for the proposed algorithm are:

- **Fog Device: -** These nodes are similar to data center servers which have all the capabilities like computation, storage, memory, and also the downlink and uplink bandwidths.

- **Sensors: -** In the simulator, there is a sensor class entity that is similar to the sensor in the Internet of Things (IoT). It primarily identifies the tuples arrival rate as well as the output characteristics of sensors.

- **Actuator: -** The class in the simulator is primarily used to perform the actions based on the tuples arrival from modules that are available in the application.

- **Tuple: -** This serves as the fundamental means of communication among the fog and a layer for data streams in the architecture as well.

- **Application: -** Based on this, the fog device's scheduling or placement of the application modules is determined.

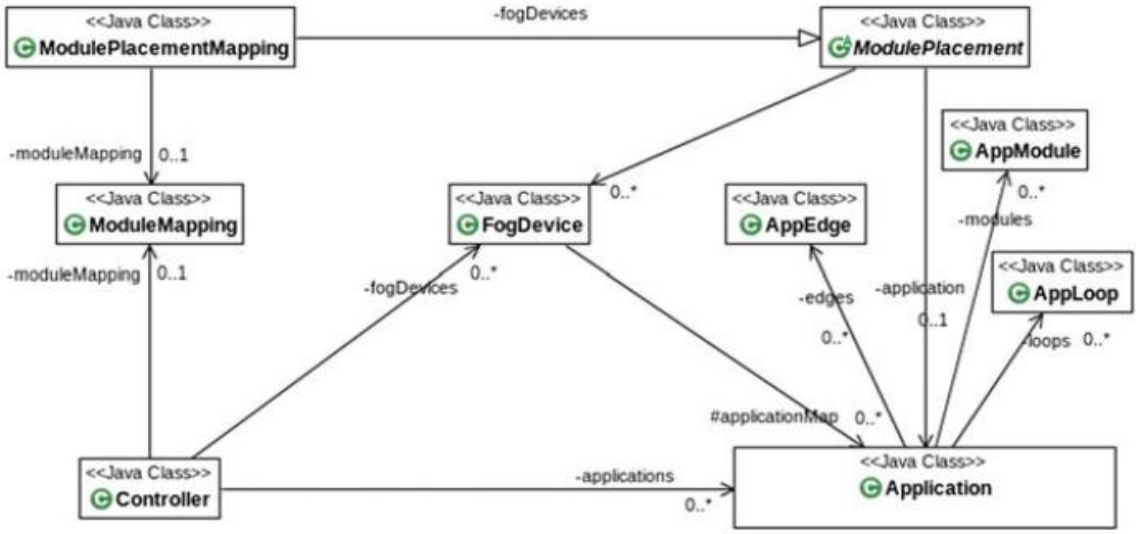

Figure 3: - iFogSim Simulator Main Classes

3.2 Cloud Analyst Simulator

It is a simulation tool that is used in the cloud. This is an extended version of CloudSim. This tool helps to decide the better load balancing and scheduling algorithms for computing . In the simulator, we can configure virtual machine memory, operating system, and placement of data centers. Load balancing algorithms are mainly used to decide the allocation of virtual machines for the requests [14] . There are some virtual machines algorithms are:

Round Robin Algorithm:

It comes under the scheduling algorithm and it's an static approach **[**14]. The requests received from the users are placed in a circular manner and for each task with some time slots. It works on the first come first serve base (FCFS). It's easy to implement, and if the virtual machines are unavailable then the request waits in the queue.

Throttled Load Balancing Algorithm:

It's a dynamic approach. There are data center controller(DCC) and virtual machine load balancer(VMLB) **[**14] are two main things in the algorithm. The requests from the IoT devices are sent to the DCC and then the VM load balancer has the status of all the virtual machines. At first, all virtual machines are available. When the data center controller requests the about the available VMs. Then the VMLB checks the table and based on the available VM, it will assign the respected virtual machine to the task.

Equally Spread Current Execution Load Balancing Algorithm:

This is also called active virtual machine load balancing. Here also virtual machine load balancer has the VM's list and also has information about the load on the servers[14]. The drawback of this algorithm is, because of the overhead caused by the DCC for updating the index table leads to a delay in providing the response to the received requests.

3.3 Proposed Computation Based Load Balancing Algorithm

The requests that are raised from the IoT devices are sent to the fog nodes. The tasks are sent to the nearest fog node if the load on that server is high and then the task is assigned

to the fog nodes based on the RAM size of the fog node. The user will receive the response after task completion.

Implemented Algorithm:-

Energy refers to -> RAM;

Instantiate fogDevices,sensors,actuators,numOfAreas,numOfCamerasPerArea,CAM_TRANSMISSION_TIME,CLOUD,evenFogDeviceRam,oddFogDeviceRam;
Step 1: Create Application with RAM assigned to picture-capture and floor-finder modules
Step 2: Loop CAMERA, picture-capture, floor-finder, PTZ_CONTROL(Actuator)
Step 3: Create cloud Fog Device And set Parent as -1 since top level Fog node
Step 3: Create proxy-server Fog Device And set Parent as Step 1 cloud id
STEP 4: Based on numOfAreas size Create Fog Devices Create a for loop
With Even Devices -> With RAM than Assigned RAM
Step A) Continue for loop since not sufficient RAM and No Camera assigned to Even Fog node
With Odd Devices -> With RAM then Assigned RAM
Step A) Since Fog Nodes RAM sufficient assign same no of cameras as numOfCamerasPerArea size

STEP 5:for loop fogDevices : While Assigned Fog Node Energy > Available Fog Node Energy
Step A : Find name of Fog Node with highest Available Energy
Step B : Find id of Fog Node with highest Available Energy
Step C : Start for loop using numOfCamerasPerArea size
Create cameras name - using Step 1 Name of Fog node and
Assign parent ID to Camera using id of Step 2 which
will create Down Fog Node cameras under highest computational RAM
Step 6: Link picture-capture to CAMERA and floor-finder to Fog Nodes
Step 7: Submit and Run the application which will print the results

4 Design Specification

Much research is performed on the load balancing algorithms to provide a better quality of service(QoS) in fog computing. From the literature survey, I understood that each proposed and designed load balancing algorithm has pros and cons. To design the model that ensures the load balancing algorithm with improved quality of service(QoS). The proposed computation based load balancing algorithm [9] have taken and also improved the algorithm. In that paper, the authors are assigning based on the energy level of the task assigned. The proposed load balancing algorithm assigns the tasks based on the computational capability if the nearest fog node load is overloaded and also before assigning the task the load on the fog nodes is also taken into consideration.

Figure 4 represents the suggested method workflow. In which IoT devices gather user requests, which are then forwarded to other fog devices for execution. Then, it will check the load on the fog nodes and then the computational capability for the requests is calculated as well if it does not meet the capability and then it assigns the task to the next high computational nodes. This approach is repeated until all the tasks are completed.

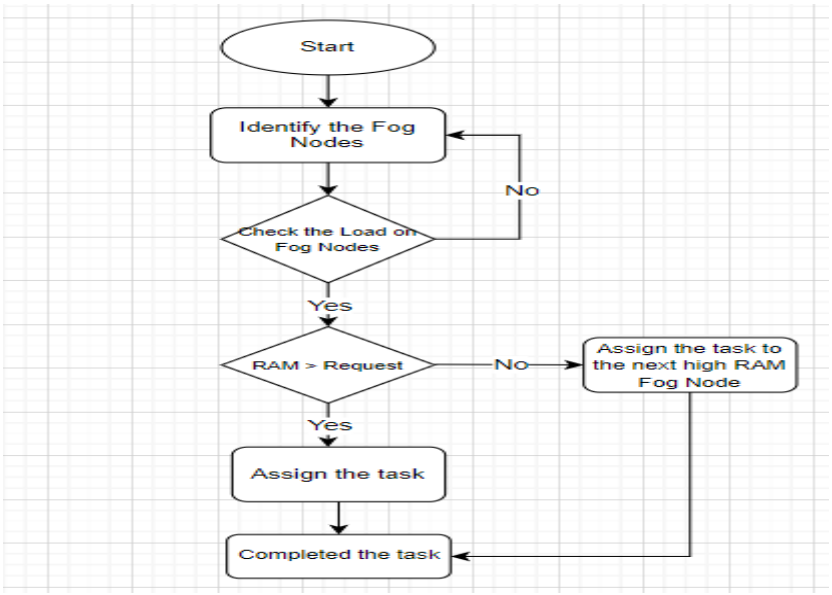

Figure 4:- Workflow for Proposed Method

4.1 Implemented Network Topology for Proposed Method

For the proposed model, I am using the iFogSim simulator in resource management for the design of a computation-based load balancing algorithm. In the suggested algorithm, I am considering the RAM size of the fog node before assigning the task. It is a three-level architecture, every IoT device is linked to fog nodes, which are then connected to the cloud. This is the implemented network topology shown in the below figure5.

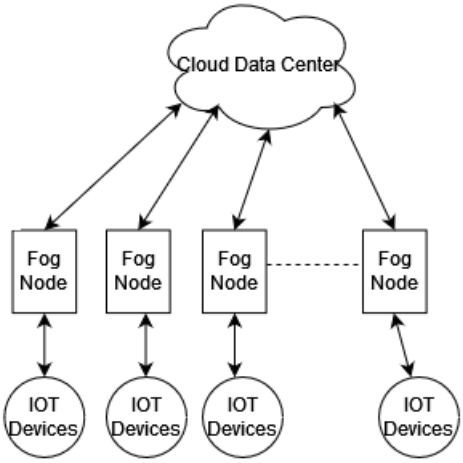

Figure 5:- Implemented Network Topology

4.2 iFogSim Simulator Configuration

The suggested architecture's main aim is to distribute tasks equally on all the fog nodes. To assign the tasks on the fog nodes the computation is taken into consideration before assigning the task. To achieve this, in iFogSim simulator has done a few configurations for the fog and cloud servers that are shown below tabular.

Parameters	Specifications
System Architecture	x86
Operating System	Linux
Virtual Machine	Xen

Table 2:- Cloud Configuration

Parameters	Specifications
Storage	100000 GB
MIPS	2800
Bandwidth	10000

Table 3:- Fog Nodes Configuration

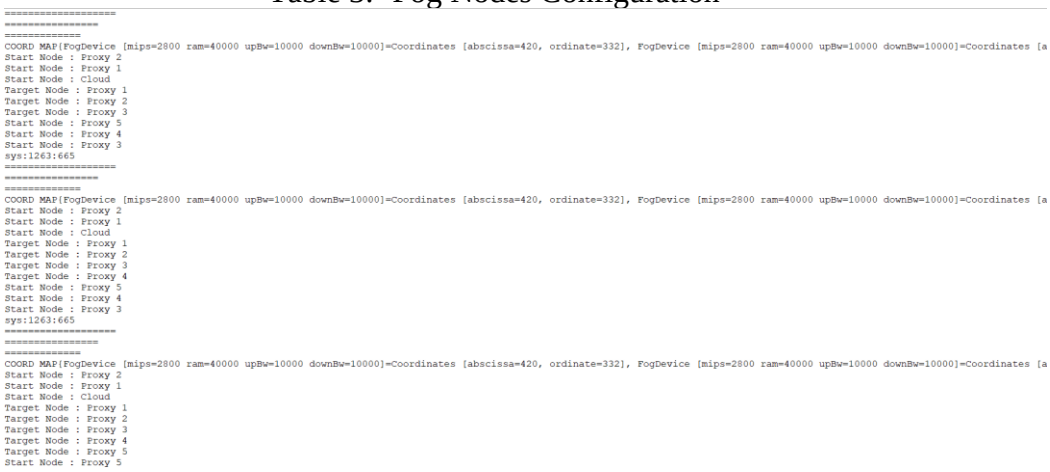

Figure 6:- Parameters Configuration in iFogSim Simulator

5 Implementation

5.1 iFogSim Simulator Implementation

Due to the costs of putting up a real cloud computing architecture scenario, for the research, I adopted the iFogSim simulator for the design of this architecture.

In the simulator, I configured the virtual machines, architecture, and operating system. The analysis and design of network topology were performed on the 64-bit windows 10 operating system with an intel i5 processor and also Eclipse IDE as well as JDK version 17 used. Due to the fact that the iFogSim was created in Java, the algorithm implementation was also created in Java using the Eclipse IDE.

The Java class which are taking part in the implementation :

- In the section of org.fog.test.perfeval, this class has a different number of other algorithms for different applications which are inbuilt in the iFogSim simulator. I designed a load balancing algorithm, when the simulation starts it initiates the architecture, virtual machine, operating systems, storage, and RAM. Then the task assigns based on the computational capability of the fog node if the load on the nearest fog node is load is overloaded.
- In this implementation org.fog.gui.example is used for the creation of network topology. By running this in the simulator, I configured the cloud virtual machine, proxy server, fog node, and IoT device parameters.
- This algorithm is implemented by dividing the fog servers into two sections even and odd fog nodes by giving the number.
- In which, the RAM is different for the fog nodes, and this helps to achieve the proposed load balancing algorithm.

5.2 Load Balancing

There is a Data center controller class which able to manage the load balancing. This is a cloud resource which able to deal the virtual machine processing queries. The VM allocation policy class was configured to work effectively and also allocates the virtual machines.

5.3 Fog Environment

The setting up of the fog environment in an iFogSim is explained in this section. This section contains the communication between the cloud layer, proxy layer, fog layer, and IoT devices. In the cloud layer, it is similar to cloud data centers which have all the computational and infrastructure resources.

5.4 Cloud Analyst Tool

Based on the proposed algorithm, I am taking the comparative analysis that exists in this software tool [14]. This simulation was performed on a 64-bit windows 11 operating system with an Intel i5 processor.

Cloud Analyst Tool:-

This is a graphical user interface tool that performs simulation and testing. This is also an extension of the CloudSim simulator. This helps the research to focus more on other characteristics like data centers and many more rather than the coding. There are main components in the configuration are,

Region:-

In the tool, the complete world map is divided into six regions. Six continents into regions i.e., North America, South America, Europe, Asia, Australia, and Africa. In these regions the main components like data center and userbase are present. These regions are shown in Figure 7.

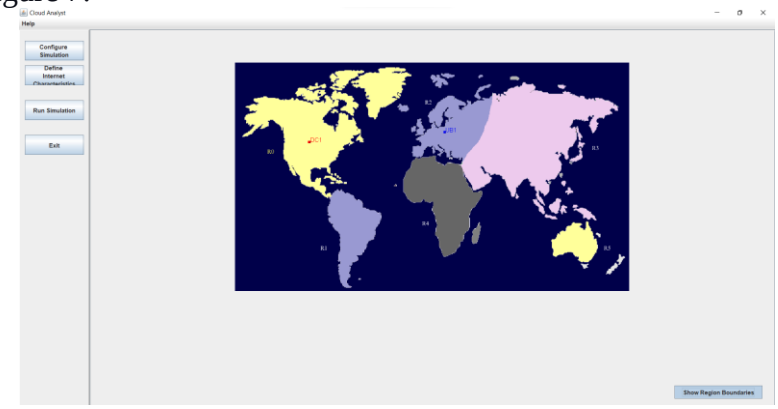

Figure 7:- Regions in Cloud Analyst Tool

Data Center:-

This is a data center controller(DCC). DCC manages all the data center operations, including cloudlets routing requests, destruction of virtual machines, creation, etc.

Virtual Machine Load Balancer(VMLB):-

The Data Center controller communicates with the VMLB for the virtual machines. The VMLB is used for the assigning of virtual machines for task completion. The cloud analyst toolkit consists of three in-built VMLB they are, Throttled, Round Robin, and Equal distribution of the current load.

Userbase:-

In a cloud analyst tool, the userbase is a collection of users who took participation in the simulation, and they are regarded as a single entity. The component's main duty is to generate traffic for the data centers like users.

Cloud Service Broker Policies:-

This is used for the managing of traffic between the user bases and data centers. The cloud analyst tool consists of inbuilt service policies they are, closet data centers, dynamically reconfigured, and optimizing the response time.

6 Evaluation

To evaluate the efficiency of the suggested computation based load balancing algorithm, different tests were conducted in the iFogSim simulator and took the comparison of cloud analyst existing algorithm with different configurations as well as a few parameters compared with the existing DCNS and Cardio Vascular health monitoring which are existed in simulator.

The network design used for this research case study for implementing the load balance method is an example of finding a vacant room in the hospital. To implement this method, a few parameters are considered which come under the improving of quality of service are,

- Energy Consumption:- This is a measure of energy consumption for the processing of the task.
- Network Cost:- Total network cost for processing the task.
- Time Taken:- The amount of time needed for the task completion.

6.1 Experiment 1 / Energy Consumption

For this experiment, I conducted a total of 4 iterations carried out to evaluate the average energy consumption of the implemented algorithm. This experiment is initiated by running the simulator environment and also creates tasks for the fog devices for processing. The results are taken from the proposed algorithm and default (DCNS) algorithm in the simulation tool. The output is shown in below (figure 10). From the outputs, I understood that the energy consumption is less by comparing with the existing algorithm in the iFogSim simulator. The demonstration of energy consumption is shown below. From the below chart, it was clear that the proposed algorithm energy consumption is less and also its efficiency in providing a better quality of service(QoS).

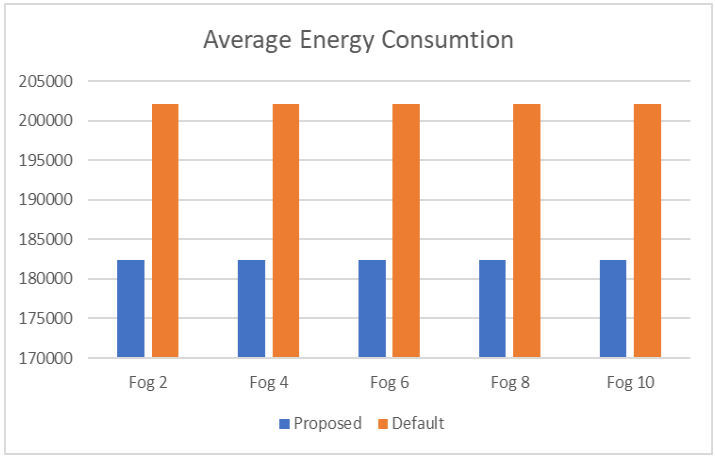

Figure 10:- Average Energy Consumption

6.2 Experiment 2 / Network Usage

During this experiment, the total consumption of the network is calculated. The network usage is measured in kilobytes/megabytes per second. The bandwidth consumption is less which means the data transfer between the from and back to the cloud is decreased. The results are taken from the proposed algorithm and also the default(DCNS) algorithm is compared. By conducting different case studies, I conclude that the suggested algorithm uses less bandwidth than by comparing with the existed algorithm in the simulator tool. The results are shown in below figure 11.

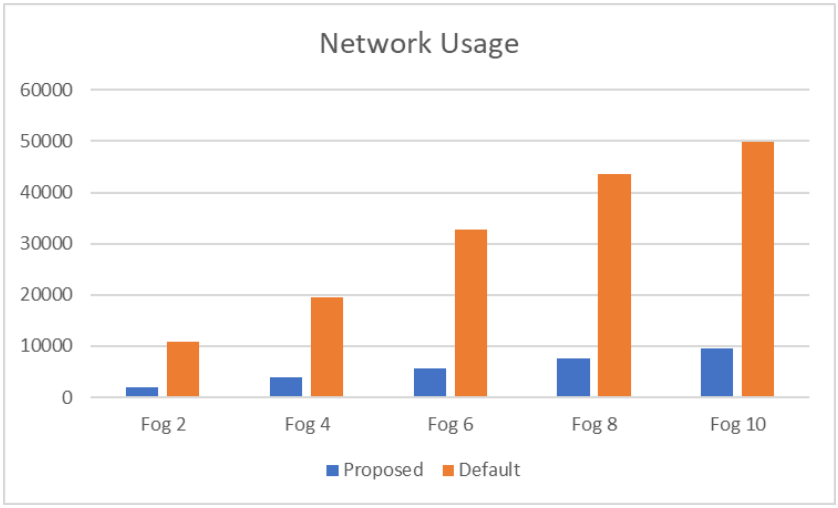

Figure 11:- The Total Network Usage

6.3 Experiment 3 / Network Cost

By performing the experiment, we are analysing the network cost for execution. The total network cost of execution is a crucial factor for the cloud user and also for the cloud providers. Hence, I took the difference between the proposed algorithm and the default (DCNS) algorithm is taken on all the fog nodes. From the results, it's evident that there is a drastic change in cost for the suggested algorithm than the existed algorithm in the simulator. This helps to provide better usage and advantages for cloud users as well as providers. The results are plotted in figure 12,

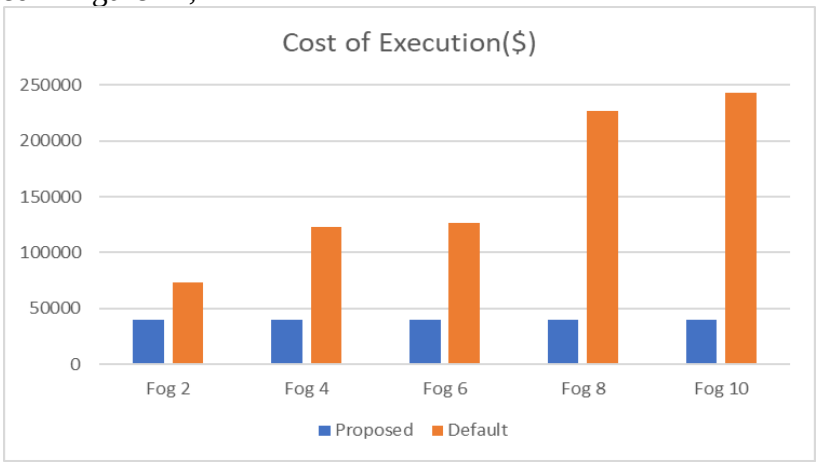

Figure 12:- The Network Cost

6.4 Experiment 4/ Task Completion time

During this experiment, the comparison of task completion is taken into consideration. By running the experiments, the proposed algorithm consumes less time for the task completion than the existing (cardiovascular health monitoring application) in the simulator. The suggested algorithm is also compared with the round-robin(figure 14), throttled (figure 15), and equally spread the current execution (figure 16) algorithms which are default algorithms in cloud analyst tool, but these existed algorithms are performing better than the proposed algorithm. The observations are plotted below in figure 13.

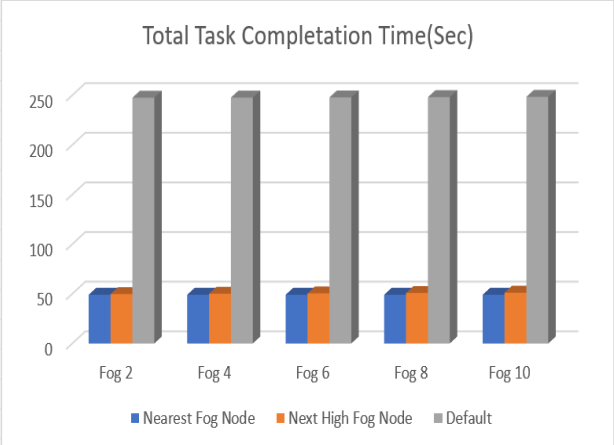

Figure 13:- Total task Completion (iFogSim)

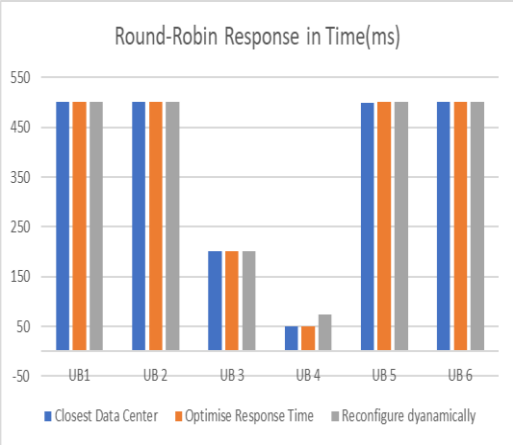

Figure 14:- Round-Robin Algorithm

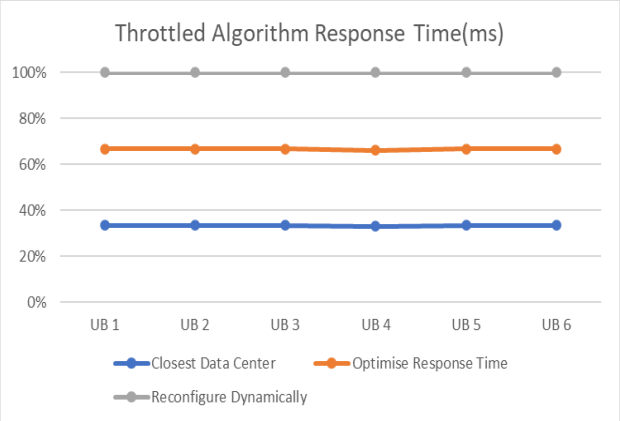

Figure 15:- Throttled Algorithm

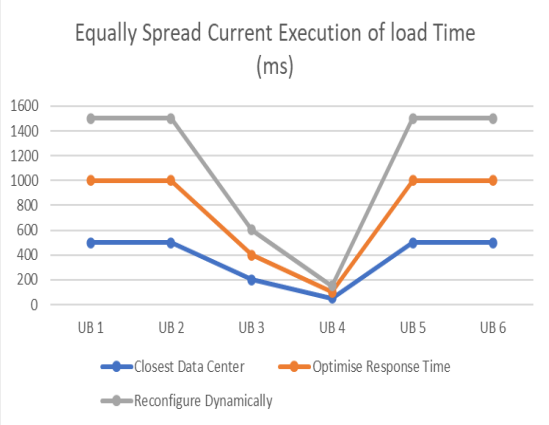

Figure 16:- Equally Spread Current Execution

6.5 Discussion

From all the above experiments, I can conclude that the proposed computation based load balancing algorithm is assigning the tasks to the fog nodes based on the RAM size and also the load on the fog devices. The experiment results show that the fog servers are not overloaded with the requests and also assign the task based on the capability of the fog devices if the load on the nearest devices is loaded. By reviewing the results, the energy consumption is reduced as shown in above figure 10. From figure 11, the bandwidth is also lower. Figure 12 plots the cost of execution. Figure 13 shows the latency is also minimized by comparing it with the default simulator algorithm.

However, by comparing latency with the cloud analyst tool the outputs are not satisfied and also compared with existing algorithms in the simulator in this the results are satisfying. As there is the quality of service (QoS) parameters, network usage, energy consumption, and cost of execution. From the outcomes, I can conclude that improving these

QoS services is achieved by less usage of resources. This suggestion will help to provide better outcomes and great usage for IoT devices.

After performing the simulation in the iFogSim simulator, the proposed and existing load balancing algorithms values are presented in the below table.

7 Conclusion and Future Work

As the usage of services and applications in the internet of things is increasing, fog computing is the fastest growing infrastructure in the area of computing that brings cloud services near to the end users. However, fog computing usage is gradually increasing, and there are challenges like improving the quality of service and also distribution of load on all the fog nodes also need to be standardized to provide better service for the users. To achieve improved services, one of the better ways is by implementing a load balancing algorithm in the fog environment. To accomplish this, I configured and proposed a computation based load balancing algorithm for assigning tasks on fog nodes. By taking the comparison with the default algorithm, the outputs suggested load balancing policy achieved the task assignment based on the computation capacity. The implemented proposed algorithm is able to achieve less usage of network usage, energy consumption, cost of execution, latency, and as well as the load on the fog nodes is not overloaded. The simulation is performed in the iFogSim simulator software where the suggested algorithm is implemented and tested on different scenarios.

With regards to the limitation, the proposed algorithm was tested on different case scenarios repeatedly and in a simulation tool. This can experiment with in the real world and tested on different cases. In future work, the energy level of the fog node can also be taken into consideration, and also the suggested method can be tested on different case studies as well as scenarios.

8 Video Link

The Video Link:- https://youtu.be/UhEYA_pv73g

References

[1] S. N. S. G. C. R. C. Y. S. B. V. E. G. B. J. L. M. V. M. A. Rajukumar Buvya, "A Manifesto for Future Generations Cloud Computing: Research Directions for the Next Decade," *ACM Computing Surveys*, vol. 51, no. 5, p. 38, NOV 2018, JCR Impact Factor: 14.324.

[2] P. S. S. S. G. Jagdeep Singh, "Fog computing: A taxonomy, systematic review, current trends and research challenges,," *Journal of Parallel and Distributed Computing,*, vol. 157, no. ISSN 0743-7315,, pp. 56-85, 2021, JCR Impact Factor: 4.542.

[3] R. K. a. G. S. a. G. D. a. J. P. P. a. G. L. a. X. Y. a. R. R. Naha, "Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions," *IEEE Access*, vol. 6, pp. 47980-48009, 2018, JCR Impact Factor: 3.476.

[4] P. S. S. S. G. Jagdeep Singh, "Fog computing: A taxonomy, systematic review, current trends and research challenges,," *Journal of Parallel and Distributed Computing,*, vol. 157, no. ISSN 0743-7315,, pp. 56-85, 2021, JCR Impact Factor: 4.542.

[5] M. a. W. L. a. B. T. a. W. A. Al-khafajiy, "Towards Fog Driven IoT Healthcare: Challenges and Framework of Fog Computing in Healthcare," in *Association for Computing Machinery*, Amman, Jordan, 2018.

[6] M. a. M. E. Haghi Kashani, "Load Balancing Algorithms in Fog Computing: A Systematic Review," *IEEE Transactions on Services Computing,* pp. 1-1, 2022,JCR Impact Factor: 11.019.

[7] Simar Preet Singh, "Effective load balancing strategy using fuzzy golden eagle optimization in fog computing environment," *Sustainable Computing: Informatics and Systems,* vol. 35, no. ISSN 2210-5379, 2022, JCR Impact Factor: 4.923.

[8] M. K. a. M. M. H. Hussein, "Efficient Task Offloading for IoT-Based Applications in Fog Computing Using Ant Colony Optimization," *IEEE Access,* vol. 9, pp. 37191-37201, 2020,JCR Impact Factor: 3.476.

[9] R. A. Mandeep Kaur, "Energy-aware load balancing in fog cloud computing,," in *Materials Today: Proceedings,*, 2020.

[10] A. U. a. A. Z. a. J. A. I. a. A. M. A. a. O. M. a. U. A. I. a. A. J. Rehman, "Dynamic Energy Efficient Resource Allocation Strategy for Load Balancing in Fog Environment," *IEEE Access,* vol. 8, pp. 199829-199839, 2020, JCR Impact Factor: 3.476.

[11] S. J. N. S. K. F. M. Rehman, Min-Min Scheduling Algorithm for Efficient Resource Distribution Using Cloud and Fog in Smart Buildings, Springer International Publishing, 2019, pp. 15--27.

[12] H. S. a. S. R. Arri, "Energy Optimization-based Optimal Trade-off Scheme for Job Scheduling in Fog Computing," in *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom),* 2021.

[13] A. V. D. S. K. R. B. Harshit Gupta, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments," June 2017.

[14] J. B. R. V. S. Rathore, "Analysis of Load Balancing Algorithms Using Cloud Analyst," in *Springer Singapore*, Singapore, 2019.