International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6623
***Article Received**: 28 February 2023 **Revised**: 20 April 2023 **Accepted**: 04 May 2023*

Improved QoS with Fog computing based on Adaptive Load Balancing Algorithm

Saurabh[1], Rajesh Kumar Dhanaraj[2]
[1]School of Computing Science and Engineering, Galgotias University,
Noida, India
KIET Group of Institutions,
Ghaziabad,U.P,India
*Email Id: sauravsharma1437@gmail.com, saurabh.it@kiet.edu
[2]School of Computing Science and Engineering, Galgotias University,
Noida, India
Email id: sangeraje@gmail.com

Abstract— As the number of sensing devices rises, traffic on the cloud servers is boosting day by day. When a device connected to the IoTwants access to data, cloud computing encourages the pairing of fog & cloud nodes to provide that information. One of the key needs in a fog-based cloud system, is efficient job scheduling to decrease the data delay and improve the QoS (Quality of Service). The researchers have used a variety of strategies to maintain the QoS criteria. However, because of the increased service delay caused by the busty traffic, job scheduling is impacted which leads to the unbalanced load on the fog environment. The proposed work uses a novel model which curates the features and working style of Genetic algorithm and the optimization algorithm with the load balancing scheduling on the fog nodes. The performance of the proposed hybrid model is contrasted with the other well-known algorithms in contrast to the fundamental benchmark optimization test functions. The proposed work displays better results in sustaining the task scheduling process when compared to the existing algorithms, which include Round Robin (RR) method, Hybrid RR, Hybrid Threshold based and Hybrid Predictive Based models, which ensures the efficacy of the proposed load balancing model to improve the quality of service in fog environment.

Keywords- Cloud Computing, Fog environment, Load balancing, QoS in cloud environment, QoS in fog computing, Resource Management.

I. INTRODUCTION

Fog computing is an extension of cloud computing that operates at the edge of the network, was developed in response to the rising demand for real-time processing of data &low latency services. Fog computing makes it possible to process and store data closer to the end devices, lowering latency and resulting in a system that is more effective and quicker to react. To ensure Quality of Service (QoS) in fog computing settings, good resource management is necessary given the increase in linked devices and applications.

A. Background:

Load balancing is an important aspect of resource management in fog computing, as it ensures that the resources are distributed efficiently among the various tasks and applications. However, conventional load balancing algorithms are not well suited to handle the dynamic and heterogeneous nature of fog computing environments [1]. This presents a significant challenge to ensuring QoS in fog computing systems.From last two decades, a lot of researchers are working on the fog computing domain[2][3]. Thousands of research articles are available to improve the performance and analyse the working of the fog environment. Still, there is a scope of the improvement in the domain to improve the QoS parameters. The paper contributes and give insight of various algorithms and models, whereas in the flow of content, the proposed algorithm and methodology is discussed for better adaptability of fog environment.

B. Problem Statement

Due to the dynamic and diverse nature of the system, ensuring QoS in fog computing settings is a significant problem. The limited resources available at the network's edge are being strained by an increase in connected devices and applications, which could result in a worsening of QoS. Conventional load balancing algorithms are not well suited in fog computing environments. These algorithms often use fixed thresholds and do not take into account the changing resource utilization and application requirements in real-time. This results in suboptimal resource allocation and reduced QoS.

The authors state that the problem they are addressing in the research paper is the lack of an effective and adaptive load balancing algorithm for improving QoS in fog computing environments. The authors aim to propose a new algorithm that can effectively balance the resources among the various tasks and applications in real-time, taking into account the changing resource utilization and application requirements.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6623
***Article Received**: 28 February 2023 **Revised**: 20 April 2023 **Accepted**: 04 May 2023*

The authors identify the following specific challenges in achieving effective load balancing in fog computing environments:

i. Heterogeneous resources: fog computing environments consist of many heterogeneous resources, including computing nodes, storage nodes, and networking components, which have varying capabilities and resource constraints.

ii. Dynamic workloads: the workload in fog computing environments can change rapidly, making it difficult to determine the optimal resource allocation in real-time.

iii. Limited bandwidth: fog computing environments are characterized by limited bandwidth, which can impact the performance of the load balancing algorithms and the overall QoS of the system.

iv. Latency requirements: fog computing systems are intended to support low latency services, which require fast and efficient load balancing algorithms to ensure that the resources are allocated and utilized optimally.

The authors aim to address these challenges by proposing a new adaptive load balancing algorithm for fog computing environments that can effectively balance the workload and improve QoS.During the study, many questions came into the mind of the researchers. These are added in the text to help the other researchers in finding the answers [4]. These questions are listed below:

Q1. What are the QoS parameters in fog computing environment?
Q2. How to improve the QoS in fog computing?
Q3. How is the load balancing done in fog computing?
Q4. What are the challenges in load balancing in fog computing?
Authors tried to figure out all possible solutions in the study and discussed the work done in the domain.

C. Objective of the Study

The goal of this work is to provide an adaptive load balancing method for environments using fog computing, which can enhance the QoS for applications and services. The goals of the study are:

- Address the challenges of load balancing in fog computing environments, including heterogeneous resources, dynamic workloads, limited bandwidth, and latency requirements.
- Develop an adaptive load balancing algorithm that considers the dynamic and heterogeneous nature of fog computing systems.
- Assessing the performance of the developed algorithm can be achieved by measuring QoS metrics like latency, energy consumption, and resource utilization.
- Compare the proposed algorithm with existing load balancing algorithms and demonstrate its superiority in improving QoS in fog computing environments.

The paper aims to contribute to the development of effective load balancing algorithms for fog computing systems, and to offer a foundation for upcoming research in this area.

II. LITERATURE SURVEY

This section thoroughly discussed about the three concern areas in fog environment. The first part is related to the fog computing basics, second part is the QoS parameters in fog environment; and third part is the load balancing in fog computing.

A. Overview of Fog Computing

Fog computing was initially suggested in the early 2010s as a reply to the growing requirement for IoT (Internet of Things) applications that need real-time processing of generated by devices at the edge of a network[5]. Traditional cloud computing systems are unable to meet the demands of these applications, as the data demands to be transmitted over the network to a centralized data center for processing, resulting in increased latency and decreased reliability. Fog computing addresses these limitations by bringing storage, computation,& networking capabilities closer to the data source[6-8]. Fog computing also enables the deployment of edge devices that can perform local processing and storage, thereby reducing the load on the cloud and improving the overall performance of the system. Figure 1 shows the layout of the fog system. The figure clearly describes the four major components: IoT devices, Fog nodes, Cloud servers, and the data processing large servers.

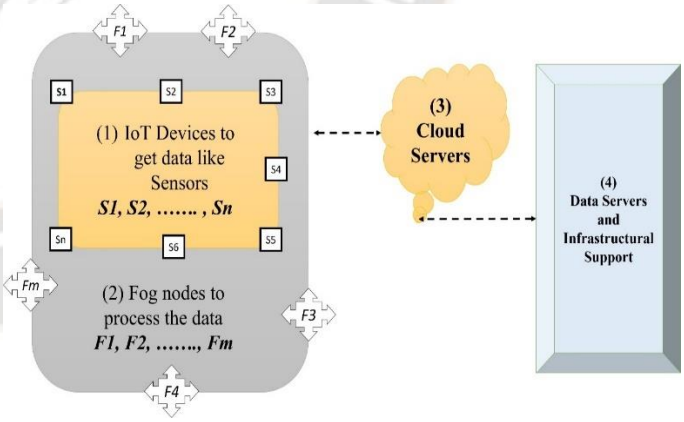

Fig 1. Layout of the fog system

a. Fog Computing Challenges

Fog computing has gained significant interest in recent years due to the growth of IoT and the increasing demand for real-time data processing. It is seen as a promising solution for a wide range of IoT applications, including smart cities, industrial automation, and autonomous vehicles [9]. Research in fog computing is focused on addressing various challenges such as

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6623
***Article Received**: 28 February 2023 **Revised**: 20 April 2023 **Accepted**: 04 May 2023*

resource management, security, scalability, and interoperability [10].

In spite of the potential advantages of fog computing, there are several challenges that must be addressed to make it a viable alternative to conventional cloud computing. Major challenges in fog computing are as follows:

i. Heterogeneity of edge devices and gateways: Edge devices and gateways can have varying hardware and software capabilities, making it difficult to develop applications and services that can run on all devices. Additionally, edge devices can have limited resources, such as processing power, memory, and storage, which can affect their ability to perform complex computing tasks. These limitations make it challenging to deploy and manage fog computing applications and services effectively [11].

ii. Limited bandwidth and connectivity of edge devices and gateways: Edge devices and gateways may be connected to the cloud through a limited bandwidth network, which can result in high latency and low throughput. This can make it difficult to transmit large amounts of data between the fog nodes of fog computing applications and services [11].

iii. Security: Edge devices and gateways can be vulnerable to cyberattacks, as they are typically deployed in public and unprotected environments. Additionally, the data transmitted between edge devices and gateways can be intercepted by malicious actors, making it important to ensure the data integrity in fog computing [11][12].

iv. Scalability: As the number of edge devices and gateways increases, it becomes more difficult to manage and coordinate their activities, which can result in decreased performance and decreased reliability of fog environment. Additionally, the dynamic nature of fog computing, where devices can come and go as needed, makes it difficult to maintain the scalability of fog computing [11][13].

v. Lack of standardization: Currently, there are no established standards for the design of fog computing, which can make it difficult to interoperate between different fog computing implementations. This can also result in increased costs for developers and organizations, as they may need to develop different implementations for different devices and platforms [11][14].

With all the above challenges in fog environment, Load Balancing is another important concern area now a days. Next section discussed about Load balancing within Fog Computing:

B. Load balancing within Fog Computing environment:

In a distributed environment, load balancing is used to disseminate the workload across multiple servers or nodes to ensure that the system is utilized efficiently, and the performance is optimized. The goal is to distribute the workload evenly so that no single node is overburdened and to prevent bottlenecks that can negatively impact performance. Whereas in fog computing, load balancing is also used to distribute the workload, but the goal is to balance the workload for effectively ensure that the system is utilized efficiently and the QoS requirements are met[15]; and leads to the main difference between load balancing systems[16]. This can provide low-latency services and improved QoS compared to a traditional distributed environment. To achieve effective load balancing in both a distributed environment and fog computing, it is important to use algorithms that can handle the dynamic and resource-constrained nature of these environments. This may involve the use of adaptive algorithms that can adjust to changing conditions, or the use of resource reservation and management strategies to ensure that the available resources are used efficiently [16][17].

Load balancing plays a crucial role in fog computing environments by distributing workloads efficiently among the available resources. The primary purpose of load balancing in fog computing is to ensure that no single resource is overburdened while others are underutilized, resulting in optimal utilization of resources and improved system performance. Load balancing in fog computing helps to:

- Improve reliability: By distributing the workload evenly, load balancing can help prevent a single resource from becoming a bottleneck, thereby increasing the reliability of the system.[18][19]
- Enhance scalability: Load balancing can dynamically allocate resources based on demand, making it easier to scale the system up or down as needed.
- Reduce latency: By directing incoming requests to the nearest available resource, load balancing can reduce the latency in data processing and response time.[18][19]

In this sub-section, the authors discuss the issue of load balancing in fog computing environments. Load balancing is a critical aspect of resource management in fog computing, as it helps to ensure that the workload is distributed evenly across the available resources, thereby improving QoS.

C. Quality of Service in Fog Computing

Quality of Service (QoS) in fog computing is determined by several factors [20], including:

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6623
***Article Received**: 28 February 2023 **Revised**: 20 April 2023 **Accepted**: 04 May 2023*

i. Latency: the time it takes for data to travel from the source to the destination.
ii. Bandwidth: The data transmission rate within a specific timeframe.
iii. Reliability: the probability that data will be delivered without errors.
iv. Availability: the probability that a service or resource will be available when needed.
v. Security: the measures taken to protect data and systems from unauthorized access, modification, or theft.
vi. Scalability: the ability of a system to handle increasing demand for services and resources.
vii. Cost: the monetary and resource costs associated with providing and maintaining a service or resource.
viii. Energy Efficiency: the ability of a system to conserve energy while meeting performance requirements.

Balancing these factors is crucial in ensuring high-quality fog computing services and applications.

S. Li et al. [35], provided a survey of various QoS-related challenges in fog computing, including bandwidth limitations, resource constraints, and latency requirements, and provides an overview of current approaches to addressing these challenges. QoS metrics and techniques for fog computing, including traffic management, resource allocation, and scheduling. The paper provided a comprehensive overview of the current research in this field and highlighted the potential of fog computing to enhance the QoS. In another research paper [36], authors addressed the issue of QoS optimization for mobile fog computing systems. The authors propose a novel mobile fog computing architecture that integrates cloud computing, edge computing, and mobile computing to provide efficient and scalable QoS optimization. The authors discussed the various challenges associated with QoS optimization in mobile fog computing systems, including limited bandwidth and computing resources, and high latency and energy consumption. They proposed a solution based on dynamic resource allocation and task offloading that takes into account both QoS requirements and resource constraints. The top view layout of the fog architecture is shown in figure 2.

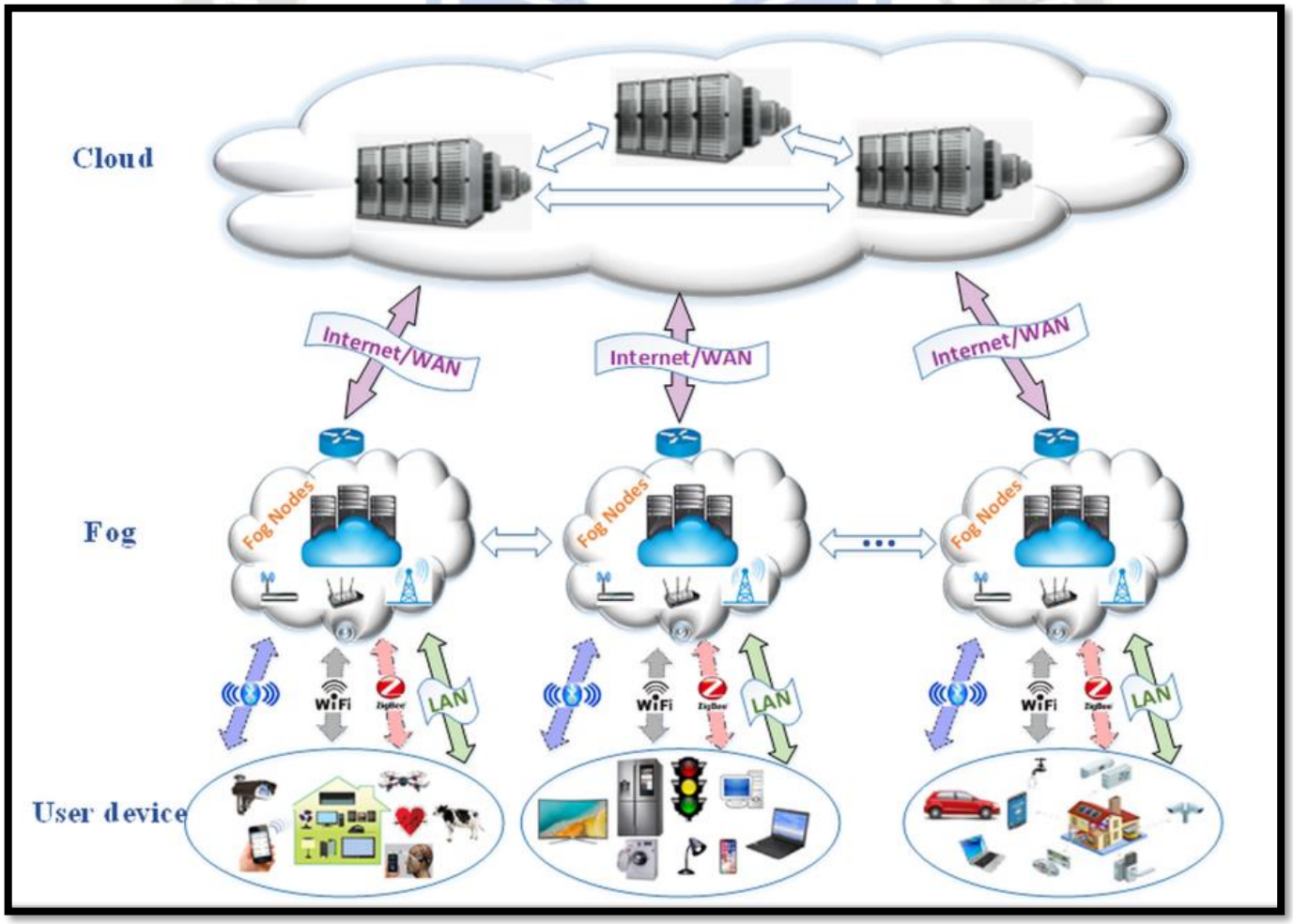

Fig 2. Fog Computing Architecture [36]

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6623
***Article Received**: 28 February 2023 **Revised**: 20 April 2023 **Accepted**: 04 May 2023*

A model is proposed where a deep reinforcement learning method for Quality of Service (QoS) aware resource allocation in the Internet of Things (IoT) and fog computing environment [37]. The authors aim to address the challenge of balancing the trade-off between resource utilization and QoS satisfaction in the IoT-fog computing ecosystem. The proposed approach uses deep reinforcement learning algorithms to dynamically allocate resources and make decisions on resource allocation to achieve a balance between resource utilization and QoS satisfaction. The results show that the proposed approach outperforms traditional methods in terms of resource utilization and QoS satisfaction. Table 1 shows the glimpse of the surveyed papers based on the working methodology and the pros-cons of the discussed models.

Load balancing can prioritize certain types of data and allocate more resources to them, thereby ensuring that certain Quality of Service (QoS) requirements are met.

Fog computing uses several techniques for load balancing [21], including:

1. Round Robin: distributing incoming requests to different nodes in a sequential manner.
2. Least Connection: directing requests to the node with the fewest active connections.
3. IP Hash: using the client IP address to determine the node that should receive the request.
4. Adaptive Load Balancing: dynamically adjusting the allocation of requests based on the current system status and performance.
5. Geographical Location: directing requests to the nearest node based on the client's geographical location.

The choice of load balancing technique depends on the specific requirements of the application and the system architecture.

TABLE 1 : LITERATURE SURVEY SUMMARY

Author	Year	Methodology	Advantage	Disadvantage
Mohammad Goudarzi [7]	2020	CWS	reduce the execution time and power consumption	high computational complexity
RedowanMahmud[6]	2019	QoS-aware placement of applications	maximize users' QoS	Power consumption
Isaac Lera [8]	2019	SPP-CN	Improves application availability	latency is difficult due to the complexity
Mahmoud M. Badawy [9]	2019	QOS-PF	maximize the quality	Delay
Liu C, Wang J, et.al, [13]	2022	MOP-IoT	Better performance compared to its counterparts in terms of various metrics	Delay
Ghobaei-Arani M, Shahidinejad A, et. al, [14]	2022	CE IoT	finding the delay reasons and work on it	computational complexity
Wu. Z et al. [15]	2020	GNN	multivariate time series is used	Only 3 parameters taken
Vaswani A, et al. [16]	2017	Survey work	Various methods discussed	Only survey paper
Habibi et al. [20]	2020	Survey work	Considered all parameters for networking	Model not implemented, only discussed the factors
Su Hu et al. [21]	2021	CNN	Textual data extracted, and the relevant content of video is predicted.	Only related to video domain
Li. M. et al. [22]	2021	TrGNN	Focuses on the traffic flow and congestion	Limited to the congestion method
Sriraghavendra, M. et al. [30]	2022	DoSP, GA	Improvement in QoS	Delay
Nair, A.R., and Tanwar S. [32]	2021	Survey paper	Discussed all QoS factors and analyzed various methods	Only survey paper
Syed Mujtibaet al. [34]	2022	Flamingo Search with GA	Fog Task scheduling Algorithm	Optimization algorithms
S. Tewari, et al. [36]	2023	Authentication methods	Used the security enhancement mechanism in paper	Delay
Jain, V. et al. [37]	2023	MDP	Improved QoS	Delay

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6623
***Article Received**: 28 February 2023 **Revised**: 20 April 2023 **Accepted**: 04 May 2023*

D. Challenges in load balancing in fog computing

Load balancing in fog computing faces several challenges [22][23], including:

1. Heterogeneity: Fog nodes may have different capabilities and resources, making it difficult to balance the workload evenly.
2. Dynamic Environment: The number and distribution of devices and nodes in the fog can change rapidly, making it difficult to maintain an efficient load balancing strategy.
3. Network Latency: Load balancing algorithms that rely on communication between nodes may introduce additional latency, affecting the overall QoS.
4. Resource Constraints: The restricted resources of fog nodes can limit the amount of processing they can perform, making it challenging to balance the workload effectively.
5. Real-time Requirements: Some fog computing applications have real-time requirements that must be met, making it difficult to balance the workload in a way that meets these requirements.
6. Security Concerns: Load balancing algorithms may be vulnerable to attack, and care must be taken to ensure that they do not introduce security vulnerabilities.

In addition to the listed ones, load balancing is the factor which influences all the parameters.

III. LOAD BALANCING IN FOG COMPUTING

The comparative analysis of existing load balancing algorithms for fog computing typically involves evaluating different algorithms based on various metrics.Some of the commonly used load balancing algorithms for fog computing include [24]:
i. Static algorithms: These algorithms use pre-defined rules or policies to distribute the workload in fog computing environments. They do not consider the dynamic nature of the system, and use pre-defined parameters, such as resource utilization and network congestion, to determine how the workload should be distributed.[24] Some common static algorithms used in fog computing include [25][26]:

- Round Robin: This algorithm distributes the workload evenly among the available resources, by assigning each new task to the next available resource. This algorithm is simple and efficient, but may not always provide optimal QoS, as it does not take into account the varying processing capabilities of the resources.[25]
- Least Connected: This algorithm assigns new tasks to the resource with the fewest active connections, ensuring that all resources are utilized evenly. This algorithm is simple and efficient, but may not always provide optimal QoS, as it does not take into account the processing capabilities of the resources.[26]
- Weighted Round Robin: This algorithm is similar to the Round Robin algorithm but takes into account the processing capabilities of the resources by assigning a weight to each resource. Tasks are then distributed to the resources based on their weight, ensuring that resources with higher processing capabilities receive more tasks.[27]

To overcome these challenges, load balancing algorithms in fog computing need to be carefully designed and optimized to manage the dynamic and resource-constrained nature of the fog environment. This may involve the use of adaptive algorithms that can adjust to changing conditions, or the use of resource reservation and management strategies to ensure that the available resources are used efficiently. Research in fog computing is focused on addressing various challenges such as resource management, security, scalability, and interoperability.

Static algorithms have the advantage of being simple and efficient, but may not always provide optimal QoS, as they do not take into account the dynamic nature of the system. In order to address these limitations, dynamic and hybrid algorithms have been proposed, which take into account the real-time information, such as resource utilization and network congestion, to improve QoS in fog computing environments.
ii. Dynamic algorithms: These algorithms use real-time information, such as resource utilization and network congestion, to distribute the workload in fog computing environments. These algorithms are more effective in ensuring QoS but may be more complex and computationally expensive [24].Some common dynamic algorithms used in fog computing include:

- Threshold-based: This algorithm monitors the resource utilization of each node and assigns tasks to the node with the lowest utilization, as long as the utilization remains - below a pre-defined threshold. If the threshold is exceeded, the task is assigned to another node. This algorithm is effective in ensuring that resources are utilized efficiently, but may not always provide optimal QoS, as it does not take into account the varying processing capabilities of the resources.[28]
- Prediction-based: This algorithm uses prediction models to estimate future resource utilization and network congestion, and assigns tasks based on the predicted values. This algorithm is more effective in ensuring optimal QoS but may be more complex and computationally expensive.[29]
- Game-theoretic: This algorithm uses game theory to model the interactions between the nodes and the tasks, and assigns tasks based on the Nash Equilibrium. This algorithm is effective in ensuring optimal QoS but may be complex to implement and computationally expensive.[30]

Dynamic algorithms are more effective in ensuring QoS in fog computing environments, as they take into account the real-time

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6623
***Article Received**: 28 February 2023 **Revised**: 20 April 2023 **Accepted**: 04 May 2023*

information, such as resource utilization and network congestion, to make decisions on task assignment. However, they may be more complex and computationally expensive compared to static algorithms. Hybrid algorithms, which combine the advantages of both static & dynamic algorithms, have been suggested to address these limitations.

iii. Hybrid algorithms: These algorithms combine the advantages of both static & dynamic algorithms, providing a balance between simplicity and effectiveness. Hybrid algorithms are devised to deal with the limitations of both types of algorithms. These algorithms aim to provide optimal QoS by combining the simplicity and efficiency of static algorithms with the real-time information-based decision making of dynamic algorithms [24]. Some common hybrid algorithms used in fog computing include:

- Hybrid Round Robin: This algorithm combines the Round Robin algorithm with dynamic information, such as resource utilization and network congestion, to determine how tasks should be assigned. The algorithm uses the Round Robin algorithm to distribute tasks evenly among the resources, but also considers the real-time information to ensure optimal QoS. [31]
- Hybrid Threshold-based: This algorithm combines the Threshold-based algorithm with dynamic information, such as resource utilization and network congestion, to determine how tasks should be assigned. The algorithm uses the Threshold-based algorithm to ensure efficient resource utilization, but also considers the real-time information to ensure optimal QoS.[32]
- Hybrid Prediction-based: This algorithm combines the Prediction-based algorithm with dynamic information, such as resource utilization and network congestion, to determine how tasks should be assigned. The algorithm uses the Prediction-based algorithm to estimate future resource utilization and network congestion, but also considers the real-time information to ensure optimal QoS.[33]

Hybrid algorithms provide a balance between the simplicity and efficiency of static algorithms and the real-time information-based decision making of dynamic algorithms, making them a suitable solution for improving QoS in fog computing environments. These algorithms aim to provide optimal QoS by combining the advantages of both static and dynamic algorithms, while minimizing their limitations. The strengths and weaknesses of these algorithms vary based on the specific requirements of the fog computing environment. For example, static algorithms are simple and efficient, but may not always provide optimal QoS. On the other hand, dynamic algorithms are more effective in ensuring QoS, but may be more complex and computationally expensive. Hybrid algorithms provide a balance between the two, but may still have limitations in terms of their ability to handle complex and dynamic workloads.

For the consideration of all the parameters, the selected algorithm is Hybrid Prediction-based algorithm. The Hybrid Prediction-based algorithm is a combination of the Prediction-based algorithm and dynamic information, such as resource utilization and network congestion, to determine how tasks should be assigned in a fog computing environment. The algorithm uses the Prediction-based algorithm to estimate future resource utilization and network congestion and combines it with real-time information to make decisions on task assignment. As an outcome, the preferred one are the Hybrid prediction-based algorithms.

The Hybrid Prediction-based algorithm works as follows [34]:

- Initialization: The algorithm starts by collecting historical data on resource utilization, network congestion, and other relevant parameters. This data is used to train prediction models that will be used to estimate future resource utilization &network congestion.
- Prediction: The prediction models are used to estimate future resource utilization & network congestion, based on the current system conditions. This information is used to determine the most appropriate resource for task assignment.
- Decision Making: The algorithm considers real-time information, such as resource utilization and network congestion, and the predicted values, to make decisions on task assignment. The task is assigned and the predicted values.
- Task Assignment: Once the task is assigned, the algorithm monitors the resource utilization and network congestion, and updates the prediction models accordingly.

The Hybrid Prediction-based algorithm aims to provide optimal QoS by combining the real-time information-based decision making of dynamic algorithms with the prediction-based approach of static algorithms [32]. This algorithm takes into account the current system conditions, as well as the predicted values, to ensure that tasks are assigned to the most appropriate resource [34]. This algorithm has been found to be effective in improving QoS in fog computing. This algorithm has been shown to be suitable for real-time applications, where the QoS requirements are high, and the system conditions are subject to frequent changes. Figure 3is the contribution to the above algorithm is discussed by Syed and Begh [34].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6623
***Article Received**: 28 February 2023 **Revised**: 20 April 2023 **Accepted**: 04 May 2023*

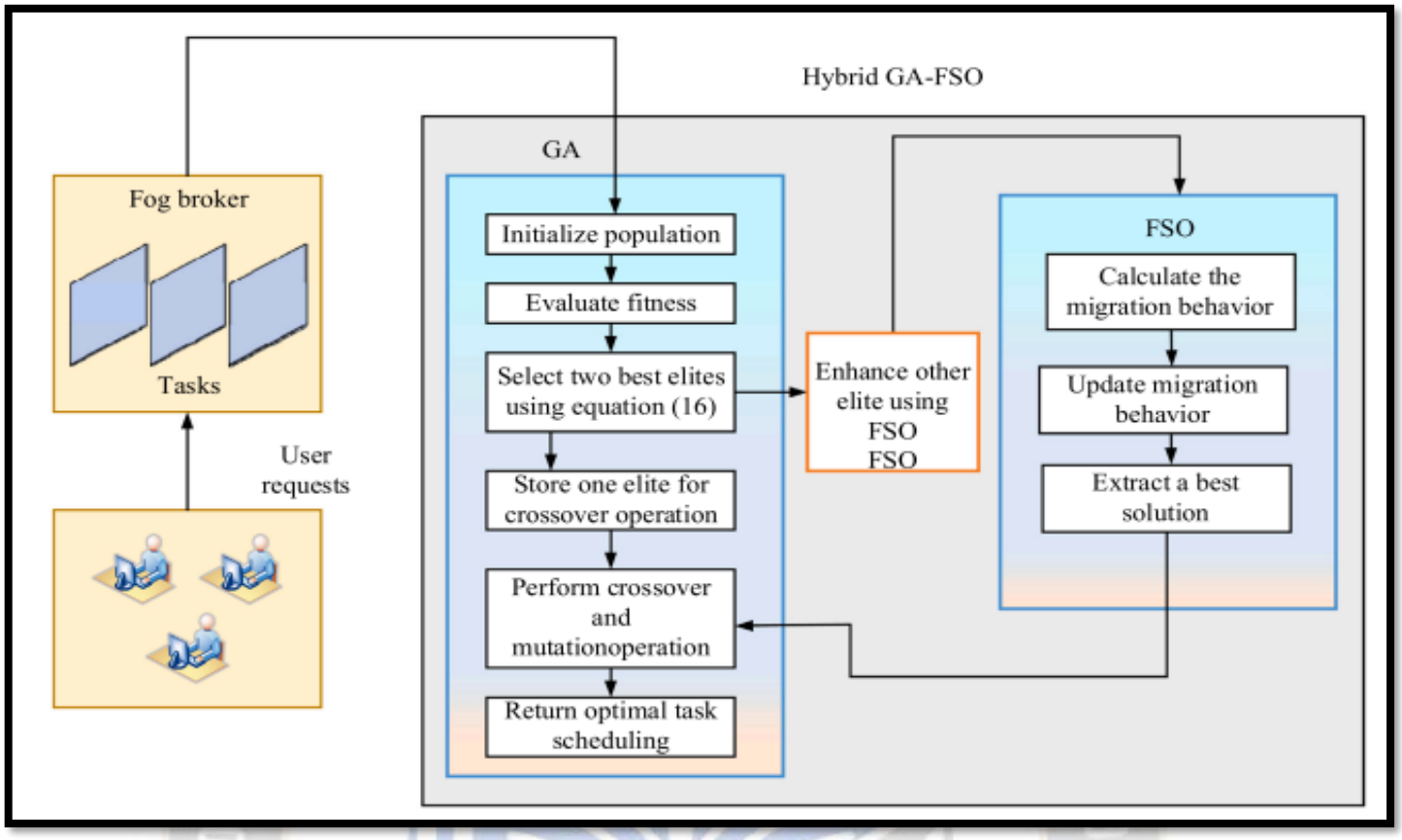

Fig 3. Task allocation model in fog environment [34]

IV. PROPOSED METHODOLOGY

As the result of the literature survey section, it has been analyzed to work on the hybrid algorithm and emphasis on the load balancing parameters, which results in the improvement in all QoS parameters. In view of Table1, where the existing models are discussed and compared, the parameters are selected and this proposed model focuses on the delay, which is disadvantage of various existing algorithms.

A. System Model

The proposed system model is based on the Hybrid Prediction-based algorithm in fog computing, which can include the following components, also shown in figure 4:

- Resource Management Unit (RMU): This component is liable for managing the available resources in the fog computing environment. The RMU collects data on resource utilization, network congestion, and other relevant parameters.
- Task Management Unit (TMU): This component is responsible for managing the tasks that need to be executed in the fog computing environment. The TMU collects data on task requirements, such as computational power and bandwidth, and forwards this information to the RMU.
- Load Balancing Unit (LBU): This component is responsible for making decisions on task assignment based on the information received from the RMU and TMU. The LBU uses the Hybrid Prediction-based algorithm to determine the most appropriate resource for task assignment, based on the present system conditions and the predicted values.
- Prediction Unit (PU): This component is responsible for estimating future resource utilization and network congestion based on the information received from the RMU. The PU uses the prediction models to estimate future resource utilization and network congestion, which is used by the LBU in decision making.
- Monitoring Unit (MU): This component is responsible for monitoring the resource utilization and network congestion and updating the prediction models accordingly. The MU collects data on resource utilization and network congestion and forwards this information to the PU.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6623
***Article Received**: 28 February 2023 **Revised**: 20 April 2023 **Accepted**: 04 May 2023*

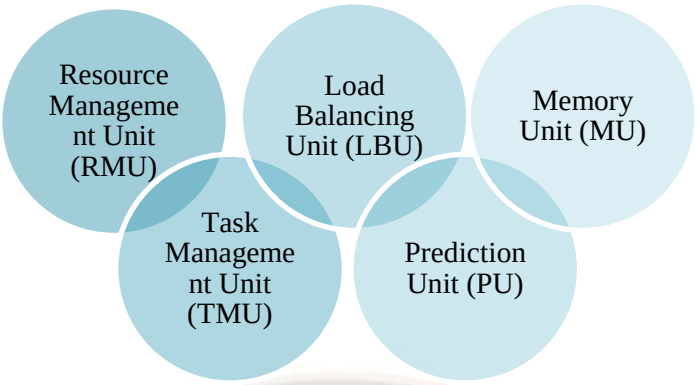

Fig 4. System Model components

The Proposed System Model for the Hybrid Prediction-based algorithm in fog computing aims to provide optimal QoS by combining the real-time information-based decision making of dynamic algorithms with the prediction-based approach of static algorithms. The different components work together to ensure that tasks are assigned to the most appropriate resource, based on the current system conditions and the predicted values. This system model provides a flexible and scalable solution for improving QoS in fog computing environments.

B. Adaptive Load Balancing Algorithm

The steps for the Adaptive Load Balancing Algorithm in a fog computing environment can be as follows:

- Resource Monitoring: The algorithm starts by monitoring the available resources in the fog computing environment, such as the CPU utilization, memory utilization, and network congestion.
- Task Submission: The next step is to receive the task requests from the clients and gather information on the task requirements, such as the computation power and bandwidth.
- Resource Allocation: Based on the information collected in the first two steps, the algorithm allocates resources to the tasks. The allocation is done using the Hybrid Prediction-based algorithm, which considers the current system conditions and the predicted values.
- Resource Utilization: The algorithm continuously monitors resource utilization and network congestion and updates the prediction models accordingly.
- Load Balancing: The algorithm adjusts the resource allocation as needed to maintain an optimal balance of resource utilization and network congestion. The algorithm makes use of both real-time information and predictions to determine the most appropriate resource for task assignment.
- Task Execution: The final step is to execute the tasks on the allocated resources and ensure that the Quality of Service (QoS) requirements are met. These steps are shown in figure 5.

Fig 5. Steps for Adaptive Load Balancing Algorithm

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6623
***Article Received**: 28 February 2023 **Revised**: 20 April 2023 **Accepted**: 04 May 2023*

The Adaptive Load Balancing Algorithm in fog computing is designed to dynamically adjust the resource allocation to improve QoS. The algorithm continuously monitors the system conditions and makes use of both real-time information and predictions to make informed decisions on task assignment. This approach provides a flexible and scalable solution for improving QoS in fog computing environments.

Basically, the two major components to work on are:

i. Monitoring Resource Utilization & Network Congestion: The algorithm continuously monitors the resource utilization and network congestion in the fog computing environment. The monitoring process is performed by the Monitoring Unit (MU), which collects data on resource utilization and network congestion and forwards this information to the Prediction Unit (PU).

ii. Updating Prediction Models: The PU uses the collected data to update the prediction models, which estimate future resource utilization and network congestion. The updated prediction models are used by the Load Balancing Unit (LBU) to make informed decisions on task assignment.

The next step of the Adaptive Load Balancing Algorithm, "Load Balancing," is crucial in maintaining optimal Quality of Service (QoS) in a fog computing environment. In this step, the algorithm adjusts the resource allocation to ensure an optimal balance of resource utilization and network congestion. The algorithm considers both real-time information and predictions to determine the most appropriate resource for task assignment. In support of the discussion, the figure 6 showed the detailed view of the system. The load balancing process can be performed in the following ways:

- Dynamic Resource Allocation: The algorithm adjusts the resource allocation in real-time based on the present system conditions and the prediction models. If the network congestion is high, the algorithm may redirect the task to another resource with less congestion.
- Predictive Resource Allocation: The algorithm predicts the future system conditions based on the historical data and makes decisions on resource allocation accordingly. The algorithm also considers the task requirements, such as computation power and bandwidth, to make informed decisions.
- Real-time Monitoring: The algorithm continuously monitors the resource utilization and network congestion and updates the prediction models accordingly. This real-time monitoring ensures that the algorithm can quickly respond to changes in system conditions and maintain an optimal balance of resource utilization and network congestion.

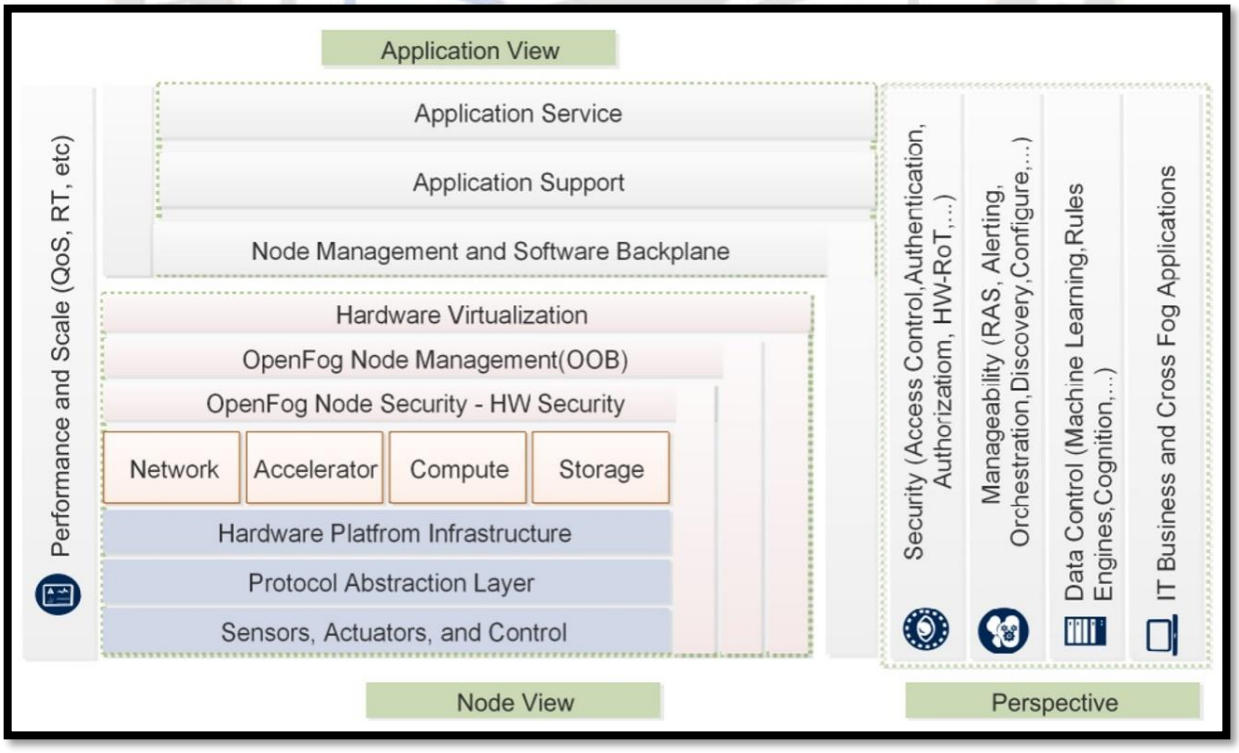

Fig 6. Architecture with load balancing problem

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6623
***Article Received**: 28 February 2023 **Revised**: 20 April 2023 **Accepted**: 04 May 2023*

The algorithm uses a combination of dynamic resource allocation, predictive resource allocation, and real-time monitoring to maintain an optimal balance of resource utilization and network congestion. To compute the results, the proposed method is to replace the weight based on *sigmoid function*, which is mathematically represented as follows:

The proposed hybrid model includes the neural network classifier, which derives condition (1) and condition (2), respectively [32].

$$\Delta S_l = -\frac{x\lambda}{r}S_n - \frac{x}{N_t}\frac{\partial C}{\partial S_n} + m\Delta S_n(t) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

$$\Delta B_n = -\frac{x}{n}\frac{\partial C}{\partial B_n} + m\Delta B_n(t) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Where $S_n$represents the sigmoid,

B_n	:	bias
n	:	layer number
λ	:	regularization parameter
x	:	learning rate
N_t	:	total number of retrieval data
m	:	momentum
t	:	represents the time, and$C$ represents

the cost function.

After replacing weight by sigmoid the fog nodes are clustered by utilizing the above conditions 1&2.

$$\tilde{J}_{\sigma n} = \sum_{l=1}^{L}\sum_{m=1}^{M}(\bar{v}_{lm})^n \frac{\|x_l - z_m'\|^2}{\bar{\sigma}_l} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

In condition (1), x_l :$l^{th}$of$d$ dimensional measured data,

q_m:m^{th}cluster center

n : constant esteem.

$\bar{\sigma}_l$: weighted mean distance in the cluster$l$,as,

$$\bar{\sigma}_l = \left\{\frac{\sum_{m=1}^{k}(\tilde{v}_{lm}^n * \bar{E}_t)\|x_l - z_m'\|^2}{\sum_{m=1}^{k}\tilde{v}_{lm}^n}\right\}^{1/2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Here, energy$\bar{E}_t$ is calculated by equation (5). The equation is used to calculate the energy consumption of a fog node at any given time t (5),

$$\bar{E}_t = \tilde{N}_T * A + \tilde{N}_R * B \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Here, $\bar{E}_t$ Signifies the energy consumed by a fog node after time t , $\tilde{N}_T$ signifies transmitted packets by a fog node, $\tilde{N}_R$ signifies the received packets by fog nodes, A and B are constants factors based upon the energy model.

$$\bar{v}_{lm} = \frac{1}{\sum_{k=1}^{q}\left(\frac{\left\|\tilde{S}_l - q_m/\bar{\sigma}_l\right\|}{\left\|\tilde{S}_l - q_k/\bar{\sigma}_l\right\|}\right)^{\frac{2}{n-1}}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

The clusters centroid [35] is computed by utilizing the condition (7),

$$z_m' = \frac{\sum_{l=1}^{L}\bar{v}_{lm}^n \cdot \tilde{S}_l}{\sum_{l=1}^{L}\bar{v}_{lm}^n} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

$$max_{lm}\left\|\bar{V}_{lm}^{(k)} - \bar{V}_{lm}^{(k+1)}\right\| < \psi \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

In condition (6), ψ is a range of 0 and 1. Rehash the steps until effective clustering got. This SNNC is represented in algorithm 1.

Algorithm 1: Sigmoid-based neural network clustering.

Input: *Fog Node set N_k*
Output: *Cluster Node*

Begin
For$k = 1toN$ *do*
Fog Node set N_k is given the coefficient v_{ij} for being a member of the cluster i
End for
Repeat
For $k = 1toN$*do*
Compute the SNNC function using condition (4)
End for
For $k = 1toN$ *do*
Compute the centroid of each cluster using condition (7)
End for
Repeat until the algorithm is met condition (8)
End

The finest clustering is acquired using this SNN. Clustering output is handled with the finest solutions. Finally, the outputs are gotten in the form of clustering. Further, the next algorithm will work on the computation of the clusters to take the query and balance the load of fog nodes. This is phase two of proposed model.

Algorithm 2: Hybrid prediction-based algorithm for fog nodes and cluster

Input: *Fog Node set N_k*
Output: *Load balanced Fog Node Set M_k*

Begin
1. ***For***$k = 1toN$ *do*
2. *Fog Node set N_k is given the coefficient v_{ij} for being a member of the cluster i*
3. *Initial the target weight with sigmoid and ReLu function;*
4. *Compute the communication cost function using threshold value T*

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6623
Article Received: 28 February 2023 Revised: 20 April 2023 Accepted: 04 May 2023

5.
6. For time ← {0 to initial computing cost}
7. Check for the request time with the threshold
8. Sum the nodes with the computing request
9. End for
10. Calculate the fog node load with the cumulative load count
11.
12. ***For*** $k = 1toN$ *do*
13. Compute the centroid of each cluster using condition (7)
14. Notify the fog servers and register the fog load limit
15.
16. After each step compute the fog nodes and monitor the computation cost
17. End for
18. Repeat until the algorithm is met condition
End

C. Performance Evaluation Metrics

For evaluating the performance of your "Adaptive Load Balancing Algorithm to Improve QoS in a Fog Computing Environment", some commonly used performance evaluation metrics include:

- Latency: measures the time taken for a task to complete from the moment it is initiated to the moment it is completed.
- Throughput: measures the number of tasks that can be completed in a given time interval.
- Resource Utilization: measures the percentage of the total available resources being used by the system at any given time.
- Accuracy: measures the correctness of the results produced by the algorithm.
- Stability: measures the robustness of the algorithm in handling varying loads and changing environments.
- Scalability: measures the ability of the algorithm to accommodate increasing loads and resources.
- Fairness: measures the distribution of resources among the tasks, ensuring that no task is given an unfair advantage or disadvantage.
- Power consumption: measures the energy consumed by the system, which is an important factor in fog computing environments where resources are limited.

In this paper, the discussed model is simulated on the system with mentioned ranges in table 2.

TABLE II: PARAMETERS AND THE VALUES USED IN SIMULATION.

Parameter	Value
fog servers count	8
IoT devices count	20
Fog node CPU frequency	2.9–4.2 GHz
IoT CPU frequency	16–84 MHz
Task size	400–800 KB
Task CPU cycle	1500–2500
Mcycle	
Bandwidth	10–20
MHz	
Noise power	–100
dBm	
Transmit power	60mW
Task deadline	2–0 s
Hidden layer	2
Type of layer	Fully
Connected	
Optimizer	Adam
Activation function	ReLu
Mini batch	64
Learning rate	0.001

V. RESULTS AND DISCUSSION

A. Simulation

This research study proposed a scheme for improving QoS Using Fog Computing Based on load balancing using hybrid predictive scheduling. The method introduced in the JAVA Cloud sim on intel i7, frequency of 2.6 Ghz, with RAM of 8GB.

B. Result Discussion

In the table, the following terms are used: Average task carrying time (Ⱥ) Average task waiting time (γ), and Average task completion time (£). All mentioned times are in ns. The demonstration is taken into consideration the existing results from [18][19][20][21], which is shown below in table 3 and figure 7.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6623
***Article Received**: 28 February 2023 **Revised**: 20 April 2023 **Accepted**: 04 May 2023*

TABLE 3. THE COMPARISON OF EXISTING ALGORITHMS WITH PROPOSED ALGORITHM.

One experiment	**Previous Algorithm [18][19][20]**	**DMOE [Su Hu 2021] [21]**	**Proposed Algorithm**
(Ⱥ)	139.4620	19.2871	15.412
γ	2.0836	2.0140	1.4915
£	251.4297	133.2391	119.212
Two experiments	**Previous Algorithm [18][19][20]**	**DMOE [Su Hu 2021] [21]**	**Proposed Algorithm**
(Ⱥ)	138.4527	20.2356	16.746
γ	2.1654	2.0451	1.625
£	250.4875	130.2491	117.198
Three experiments	**Previous Algorithm [18][19][20]**	**DMOE [Su Hu 2021] [21]**	**Proposed Algorithm**
(Ⱥ)	139.4578	19.2045	15.962
γ	2.24589	1.9201	1.546
£	253.4214	129.5017	115.966

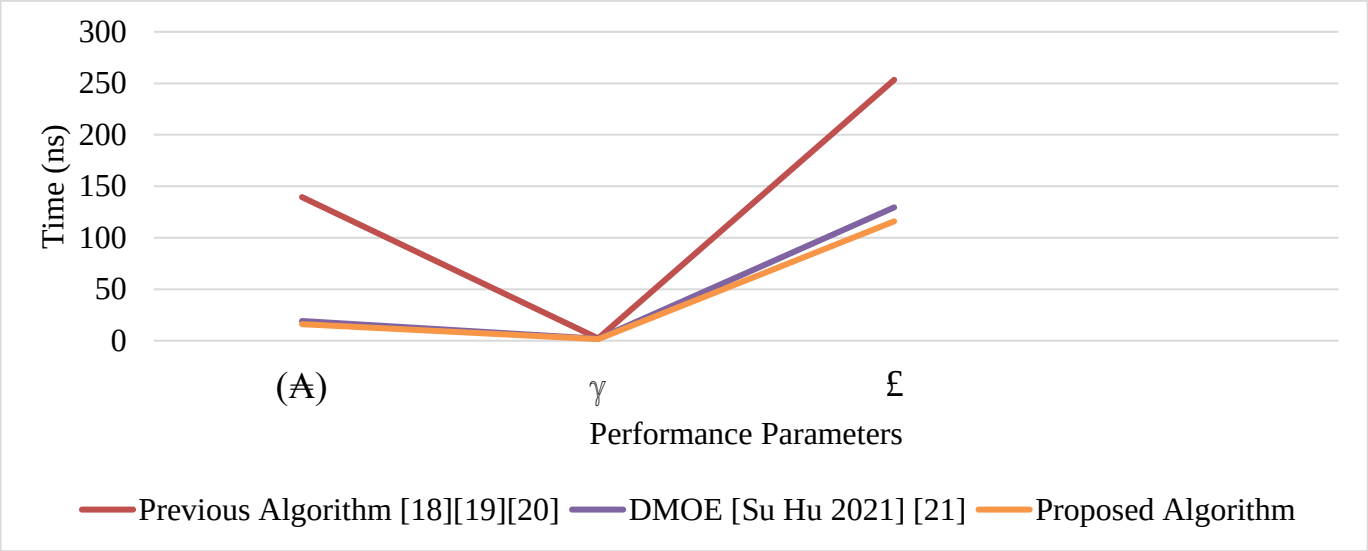

Fig 7: Performance parameters for 20 network nodes on various algorithm

C. Comparison with Existing Methods

To show the results of the Adaptive Load Balancing Algorithm in a fog computing environment, the following parameters can be used on the graph axes:

X-axis: Time or number of tasks processed. This parameter represents the progression of the algorithm over time, or the number of tasks processed by the system.

Y-axis: This parameter represents the level of resource utilization or network congestion in the system with various tasks.

Performance Metric: The performance of the Adaptive Load Balancing Algorithm can be shown as a line graph, where the line represents the value of the performance metric over time or the number of tasks processed. Common performance metrics for load balancing algorithms include Average task carrying time (Ⱥ) Average task waiting time (γ), and Average task completion time (£). These parameters can be used to show the effectiveness of the Adaptive Load Balancing Algorithm in improving QoS in a fog computing environment. The graph can demonstrate how the algorithm dynamically adjusts the resource allocation to maintain an optimal balance of resource utilization and network congestion, which results in improved QoS. The graphs shown in figure 8 and figure 9 are the drawn in comparison to the H-RR [18], H-TB[19], H-PB[20] and the HFSGA [21] existing algorithms in contrast to the proposed hybrid algorithm to compute the average communication cost and average computation cost. To compute the cost factor, the considered number of tasks are gradually increasing on the system simulated on the Cloud sim from 50, 100, 150 and 200 tasks. The summary table is shown as Table 4 and Table 5.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6623
***Article Received**: 28 February 2023 **Revised**: 20 April 2023 **Accepted**: 04 May 2023*

TABLE IV: COMMUNICATION COST ON VARIOUS NUMBER OF TASKS

Number of Tasks	H-RR	H-TB	H-PB	HFSGA	Proposed
50	10.4	9.8	7.8	8.8	6.5
100	14.8	14.2	13.11	14.18	10.48
150	18.24	17.62	17.14	17.22	14.28
200	22.86	21.71	20.42	20.19	17.14

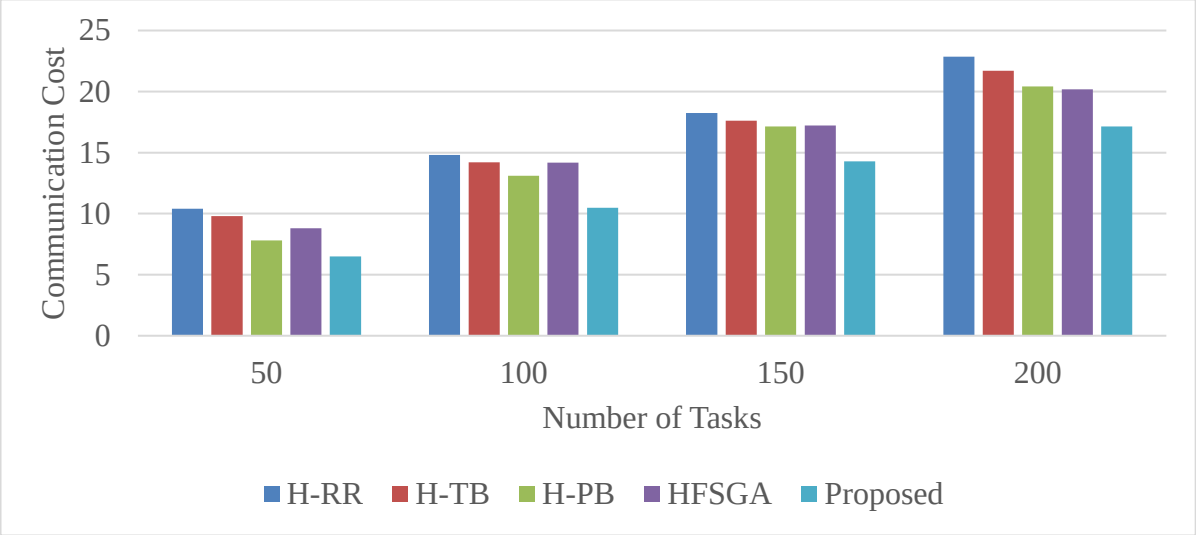

Fig 8. Visualization of communication cost with respect to tasks count

TABLE V: COMPUTATION COST ON VARIOUS NUMBER OF TASKS

Number of Tasks	H-RR	H-TB	H-PB	HFSGA	Proposed
50	187	181	175	177.8	164
100	395	372	359	363	334
150	596	571	543	548	516
200	823	786	741	723	689

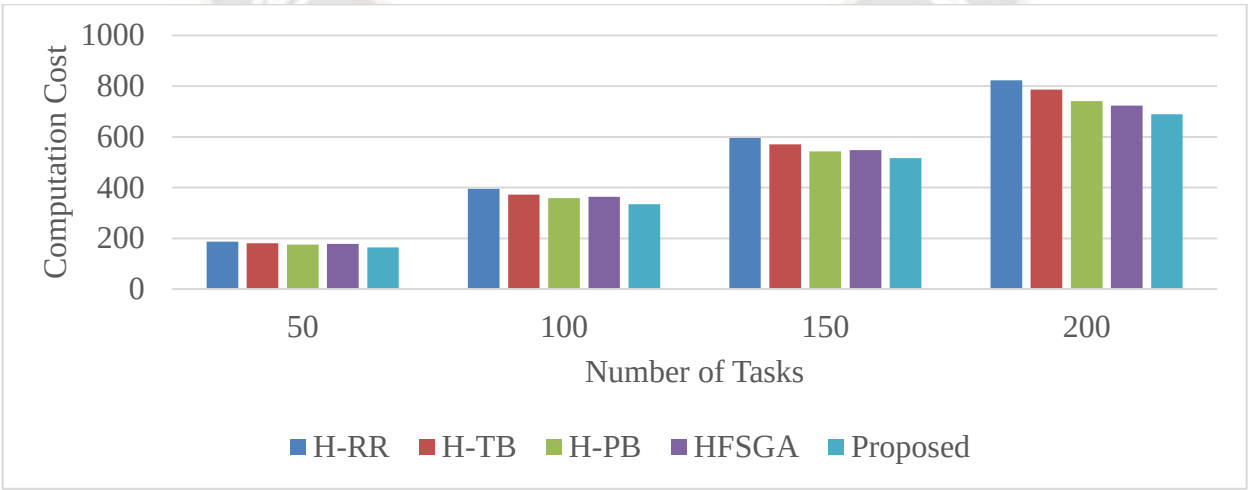

Fig 9. Visualization of computation cost with respect to tasks count

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6623
***Article Received**: 28 February 2023 **Revised**: 20 April 2023 **Accepted**: 04 May 2023*

The results shown in the above tables and figures justify the role of hybrid algorithms in the load balancing and improve the QoS of fog environment. The outcome showed reduced communication cost and the computation cost of the system.

VI. Conclusion

The problem addressed in this research paper is the need for effective load balancing in fog computing environments to ensure Quality of Service (QoS) for applications and services. Conventional load balancing algorithms are not well suited to handle the dynamic and heterogeneous nature of fog computing systems, leading to resource imbalances and poor QoS. The proposed approach shown in the paper is hybrid algorithm for load balancing in heterogenous servers. The implementation of the proposed algorithm and compared with the existing models and implemented algorithms which shows the better performance of the proposed algorithm. The parameters taken to compare are Average task carrying time (A) Average task waiting time (γ), and Average task completion time (£), which leads to the load balancing and improve other QoS parameters in terms of latency, reaction time, energy consumption, network traffic, bandwidth, transmission rate, and throughput in simulations. Further, the same algorithm can be applied to the various factors on fog nodes where mobility is introduced. Many ensuing algorithms can come with hybrid models, but the need is to work on one approach which is suitable for static as well as dynamic topology of fog servers and fog nodes. As of now, as demonstrated and discussed in the paper, the proposed one outperforms in terms of all parameters.

References

[1] Omoniwa, Babatunji, Riaz Hussain, Muhammad Awais Javed, Safdar Hussain Bouk, and Shahzad A. Malik. "Fog/edge computing based IoT (FECIoT): Architecture, applications, and research issues." *IEEE Internet of Things Journal* 6, no. 3 (2018): 4118-4149.

[2] Brogi, Antonio, and Stefano Forti. "QoS-aware deployment of IoT applications through the fog." *IEEE Internet of Things Journal* 4, no. 5 (2017): 1185-1192.

[3] Ni, Jianbing, Kuan Zhang, Xiaodong Lin, and Xuemin Shen. "Securing fog computing for internet of things applications: Challenges and solutions." *IEEE Communications Surveys & Tutorials* 20, no. 1 (2017): 601-628.

[4] Hussein, Mohamed K., and Mohamed H. Mousa. "Efficient task offloading for iot-based applications in fog computing using ant colony optimization." *IEEE Access* 8 (2020): 37191-37201.

[5] Nashaat, Heba, Eman Ahmed, and Rawya Rizk. "IoT Application Placement Algorithm Based on Multi-Dimensional QoS Prioritization Model in Fog Computing Environment." *IEEE Access* 8 (2020): 111253-111264.

[6] Mahmud, Redowan, Satish Narayana Srirama, Kotagiri Ramamohanarao, and Rajkumar Buyya. "Quality of Service (QoS)-aware placement of applications in Fog computing environments." *Journal of Parallel and Distributed Computing* 132 (2019): 190-203.

[7] Goudarzi, Mohammad, Huaming Wu, Marimuthu S. Palaniswami, and Rajkumar Buyya. "An application placement technique for concurrent IoT applications in edge and fog computing environments." *IEEE Transactions on Mobile Computing* (2020).

[8] Lera, Isaac, Carlos Guerrero, and Carlos Juiz. "Availability-aware service placement policy in fog computing based on graph partitions." *IEEE Internet of Things Journal* 6, no. 2 (2018): 3641-3651.

[9] Badawy, Mahmoud M., Zainab H. Ali, and Hesham A. Ali. "Qos provisioning framework for service-oriented internet of things (iot)." *Cluster Computing* (2019): 1-17.

[10] Jamil, Bushra, Mohammad Shojafar, Israr Ahmed, Atta Ullah, Kashif Munir, and Humaira Ijaz. "A job scheduling algorithm for delay and performance optimization in fog computing." Concurrency and Computation: Practice and Experience 32, no. 7 (2020): e5581.

[11] Deepa, N., and P. Pandiaraja. "E health care data privacy preserving efficient file retrieval from the cloud service provider using attribute based file encryption." Journal of Ambient Intelligence and Humanized Computing (2020): 1-11.

[12] Chen, Huankai, and Frank Z. Wang. "Spark on entropy: A reliable & efficient scheduler for low-latency parallel jobs in heterogeneous cloud" In 2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops), pp. 708-713, IEEE, 2015.

[13] Chen, W., Chen, L., Xie, Y., Cao, W., Gao, Y., Feng, X.: Multi-rangeattentive bicomponent graph convolutional network for traffic forecasting.In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34,pp. 3529–3536 (2020)

[14] Ghobaei-Arani M, Shahidinejad A. A cost-efficient IoT service placement approach using whale optimization algorithm in fog computing environment. Expert Systems with Applications. 2022 Aug 15;200:117012.

[15] Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., Zhang, C.: Connectingthe dots: Multivariate time series forecasting with graph neural networks.In: Proceedings of the 26th ACM SIGKDD International Conference onKnowledge Discovery & Data Mining, pp. 753–763 (2020)

[16] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez,A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances inneural information processing systems 30 (2017)

[17] Du, S., Li, T., Gong, X., Horng, S.-J.: A hybrid method for traffic flow forecasting using multimodal deep learning. arXiv preprint arXiv:1803.02099(2018)

[18] C. Wang, S. Li, T. Cheng, et al., A construction of smart city evaluation system based on cloud computing platform, Evol. Intell. 13 (10) (2019) 119–129.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6623
***Article Received**: 28 February 2023 **Revised**: 20 April 2023 **Accepted**: 04 May 2023*

[19] K. Lee, K. Kim, A performance evaluation of a geo-spatial image processing service based on open source paas cloud computing using cloud foundry on openstack, Remote Sens. 10 (8) (2018) 1274.

[20] Habibi, Pooyan, Mohammad Farhoudi, Sepehr Kazemian, Siavash Khorsandi, and Alberto Leon-Garcia. "Fog computing: a comprehensive architectural survey." IEEE Access 8 (2020): 69105-69133.

[21] Su Hu, Yinhao Xiao, "Design of cloud computing task offloading algorithm based on dynamic multi-objective evolution", Future Generation Computer Systems, Volume 122, 2021, Pages 144-148, ISSN 0167-739X, https://doi.org/10.1016/j.future.2021.04.002.

[22] Li, M., Tong, P., Li, M., Jin, Z., Huang, J., Hua, X.-S.: Traffic flow prediction with vehicle trajectories. In: Proceedings of the AAAI Conferenceon Artificial Intelligence, vol. 35, pp. 294–302 (2021) Springer Nature 2021

[23] Zheng, C., Fan, X., Wang, C., Qi, J.: Gman: A graph multi-attentionnetwork for traffic prediction. In: Proceedings of the AAAI Conference onArtificial Intelligence, vol. 34, pp. 1234–1241 (2020)

[24] Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprintarXiv:1709.04875 (2017)

[25] Tian, C., Chan, W.K.: Spatial-temporal attention wavenet: A deeplearning framework for traffic prediction considering spatial-temporaldependencies. IET Intelligent Transport Systems 15(4), 549–561 (2021)

[26] Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatialtemporal graph convolutional networks for traffic flow forecasting. In:Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp.922–929 (2019)

[27] Kong, X., Zhang, J., Wei, X., Xing, W., Lu, W.: Adaptive spatial-temporalgraph attention networks for traffic flow forecasting. Applied Intelligence52(4), 4300–4316 (2022)

[28] Oreshkin, B.N., Amini, A., Coyle, L., Coates, M.: Fc-gaga: Fully connected gated graph architecture for spatio-temporal traffic forecasting. In:Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp.9233–9241 (2021)

[29] Pan, Z., Zhang, W., Liang, Y., Zhang, W., Zheng, Y.: Spatio-temporalmeta learning for urban traffic prediction. IEEE Transactions on Knowledge and Data Engineering PP(99), 1–1 (2020)

[30] Sriraghavendra, M., Chawla, P., Wu, H., Gill, S.S., Buyya, R. (2022). DoSP: A Deadline-Aware Dynamic Service Placement Algorithm for Workflow-Oriented IoT Applications in Fog-Cloud Computing Environments. In: Tiwari, R., Mittal, M., Goyal, L.M. (eds) Energy Conservation Solutions for Fog-Edge Computing Paradigms. Lecture Notes on Data Engineering and Communications Technologies, vol 74. Springer, Singapore. https://doi.org/10.1007/978-981-16-3448-2_2

[31] X. Huang, J. Wang, Y. Lan, C. Jiang, and X. Yuan, "MD-GCN: A Multi-Scale Temporal Dual Graph Convolution Network for Traffic Flow Prediction," Sensors, vol. 23, no. 2, p. 841, Jan. 2023, doi: 10.3390/s23020841. [Online]. Available: http://dx.doi.org/10.3390/s23020841

[32] Nair, A.R., Tanwar, S. (2021). Fog Computing Architectures and Frameworks for Healthcare 4.0. In: Tanwar, S. (eds) Fog Computing for Healthcare 4.0 Environments. Signals and Communication Technology. Springer, Cham. https://doi.org/10.1007/978-3-030-46197-3_3

[33] Deng, R., Lu, R., Lai, C., & Luan, T. H. (2015). Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing. In 2015 IEEE International Conference on Communications (ICC) (pp. 3909–3914). Piscataway, NJ: IEEE.

[34] Syed Mujtiba Hussain, Gh Rasool Begh, "Hybrid heuristic algorithm for cost-efficient QoS aware task scheduling in fog–cloud environment", Journal of Computational Science, Volume 64, 2022, 101828, ISSN 1877-7503, https://doi.org/10.1016/j.jocs.2022.101828.

[35] M. Mukherjee, L. Shu and D. Wang, "Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges," in IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 1826-1857, thirdquarter 2018, doi: 10.1109/COMST.2018.2814571.

[36] Suchita Tewari, Naveen Tewari, Mukesh Joshi, Utilizing Technology and Management of Fog Computing, Cyber Technologies and Emerging Sciences, 10.1007/978-981-19-2538-2_49, (477-483), (2023).

[37] Jain, V., Kumar, B. QoS-Aware Task Offloading in Fog Environment Using Multi-agent Deep Reinforcement Learning. J Netw Syst Manage 31, 7 (2023). https://doi.org/10.1007/s10922-022-09696-y