Optimizing Resource Allocation and Load Distribution in Fog-Computing for IoT: A Hybrid Approach for Enhanced Performance

Dinesh Kumar[1], Bhavna Sharma[2]

[1,2]Department of Computer Science Engineering, JECRC University, Jaipur, Rajasthan, India
EMAIL: bhardwaj.d2009@gmail.com, bhavna.sharma@jecru.edu.in

Abstract:
Efficient resource allocation and load balancing in fog computing is critical for many IoT application systems. The purpose of this paper is to address challenges of load balancing, bottlenecking, overloading, delaying and distributing the load in fog-based IoT. The study presents a hybrid strategy combining Fog-IoT load balancing resource allocation method using Least Connection and Weighted Round-Robin. The analysis assesses the various types of data transmission using various evaluation metrics including throughput, latency, response time, queuing time, speed and required time. The findings show the methods maintains a reduction of system latency, response time and while improving channel utilization and speed.

Keywords: IoT, Round-Robin, load distribution, QoS, resource allocation etc.

1. Introduction:

As applications of IoT rapidly grow, the reliability as well as efficiency of computing systems is greatly in demand. Fog computing is a potential solution for centralized cloud computing. It promises to aid in providing computing resources close to the edge of the network. Being close by allows systems to process data faster and with less latency. Nonetheless, one of the major challenges in IoT area is the effective use of resources and load balancing in a fog-based area. The primary objective of this study is to optimize the distribution of network load in the fog-based Internet of Things (IoT) through the use of Fog-IoT load balancing resource allocation method, least connection and weighted round robin.

1.1. Research Questions: The following research questions are formed:

1.1.1. How can a hybrid approach combining Fog-IoT load balancing with least connection and weighted round-robin algorithms optimize network load distribution?
1.1.2. How it can improve system performance in fog-based IoT environments?
1.1.3. What impact does this optimization have on system performance metrics such as throughput, latency, response time, energy efficiency, bandwidth utilization and, scalability?
The questions of study focus on:

- The aim of optimization is to distribute the load of the network.
- A hybrid method was developed in this paper which uses Fog-IoT load-balancing, least connection, and weighted round robin algorithms.
- The performance measures include throughput, latency, response time, energy efficiency, and also scalability.

The goal of the research paper is to find out various sorts of Data Communication by Evaluation. Further, we analyse the Key Performance Indicator (KPI) for different modelling. The designed fog framework has a privacy and security-aware layer that executes secure task allocation, authentication, and intrusion and detection to further protect the IoT setting.

2. A Framework for an IoT Platform Based on Fog Computing

Fog computing refers to a decentralized computing infrastructure that distributes data, computing, storage, and applications between the data source or end-user and the cloud. Fog computing brings the possibilities of the cloud closer to the end user or consumer. The suggested fog computing architecture or fog-based IoT platform is shown in Figure 1. The three layers of the architecture can be named cloud, fog nodes and IoT devices. The time sensitivity of an Internet of Things application determines if intelligent control and decision-making are needed at each level.

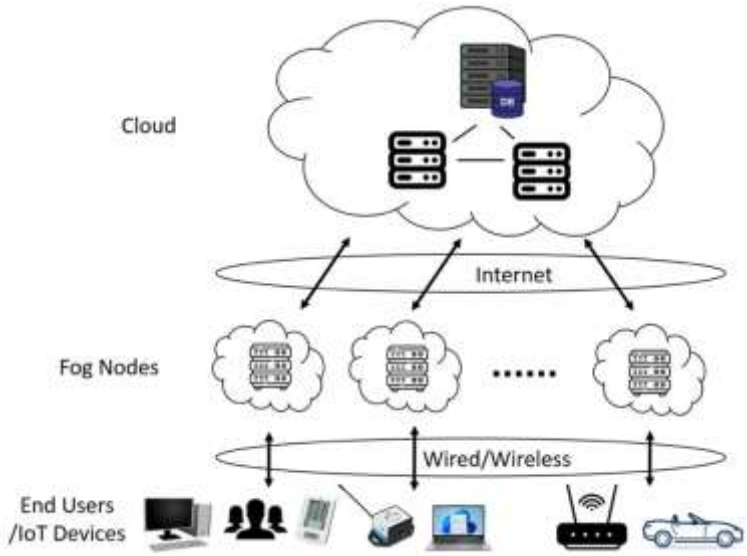

Figure 1:The fog computing system framework (Tang, S. (2023).

Bottom Layer: Many IoT devices, including robots, intelligent safety, smartphones, wearable technology, smart watches, glasses with sensors, laptops, and autonomous vehicles make up the end user layer, which is the lowest layer. While some of these devices might be capable of computing, others might just be able to gather raw data by sending it to the higher level for data processing and storage after intelligently perceiving objects or events.
Centre Layer: A collection of fog nodes, including routers as well as gateways, switches, connection points, base stations, and fog servers, make up the center layer, known as the fog layer. Independent devices known as fog nodes compute, send, and temporarily store the data produced by Internet of Things devices. A fog server uses the data to decide what to do. Typically, fog servers are connected to other fog devices. Fog nodes are connected to internet of things (IoT) devices to offer intelligent services, and they can be placed stationary or on vehicles that are moving.
The cloud layer, which is the topmost layer, is made up of several storage units and servers devices with strong processing and storage capacities that enable the provision of intelligent application services. Large data storage and a variety of computational analyses are supported by this layer. Fog computing does not, however, manage all computation and storage via the cloud, in contrast to the conventional cloud computing architecture. The fog nodes themselves possess the necessary processing and storage power. The processing activities among fog nodes and the cloud can be effectively handled and scheduled using a few control mechanisms, depending on the data processing load along with quality of service (QoS) prerequisites. This will increase the rate at which the system's resources are employed.

3. Literature Review:

One important topic is the balancing of fog computing load (Sulimani et al., 2024). Efficiently assigning various activities to fog nodes is resource allotment which ensures optimal utilization of fog resources. The purpose is to reduce bandwidth consumption and delays in service requests, and provide real-time applications support. At times fog resources in clusters are not optimally exploited. If any tasks cannot be allocated to any available resources, the tasks will go to the cloud and may delay it. A robust allocation strategy for distributing load among fog nodes is essential to enhanced efficiency of allocation of fog nodes (Ogundoyin and Kamil). The networking system can solve the issue of load balancing by several algorithms. A technique utilized in scheduling is round robin (Ali and Alubady 2023). Other methods are dynamic such as the adaptive model Wang and Lu (2022). To minimise the amount of work sent to the cloud, there is a need to apply various balancing algorithms that use the fog resources in an efficient manner (Ali and Alubady, 2023). The simplicity of the round-robin algorithm allows it to be quite a popular load balancer. According to Wang et al. (2016), the sequential execution of client requests on the cloud servers, starting at the beginning of the ordered list, is one of the round robin algorithm's critical functions for efficient load balancing. Upon receiving a client request, the client request is started again from the first node after it is assigned to the last server in the list. According to a comparative study between processor scheduling algorithms in Figure 62, the round robin algorithm is easier to implement as compared to all other load balancing algorithms. The various loading requests at the server cannot be balanced appropriately (Ghosh and Banerjee, 2018). The various processing intervals' priorities are established based on time measures parameters. According to Xu et al. (2016), the round robin method is pertinent to many time factors such as time quantization, burst time and so on. The burst time represents the total time needed to process a particular service request or a task in a system. The time quantum is the total time that a service will take to reach on a virtual machine. According to Choudhary and Kothari, 2018, researchers discovered many limitations of static round-robin technique. Some disadvantages impact the system by increasing its waiting time, response time, and context switch count. Under

other circumstances, the throughput of the configuration can also drop. Moreover, the quantum time exhibits a spaghetti-like behavior in the normal round robin scheduling algorithm.
According to Shafiq et al. (2022), many well-known load balancing techniques can be applied to improve cloud computing. Based on the operating environment, there are 3 types of algorithms- static algorithms, dynamic algorithms, and nature-inspired algorithms.
The load management and resource allocation play an important role for achieving better system performance of the fog computing environment in IoT applications. Many different approaches were proposed to counter these (centralized, distributed load balancing algorithms). Most of the time, to split the load between the available servers the least connection algorithm is used. The weighted round robin algorithm allocates server weights according to their load and capacity. According to Zhang et al. (2018), the authors proposed a fog calculating surrounding the hybrid load balancing approach based on the least of connections and the weighted rounded robin algorithm. In terms of system performance and resource utilization, the hybrid technique gave better results than either of the algorithms. Wang et al. (2019) took on the challenges of fog computing. The paper investigates load balancing algorithms for improving fog computing system performance. This paper is one like the earlier one. It also stresses the need for resource allocation. Most importantly, allocating resources efficiently improves the efficiency of any system.

4. Proposed framework:

A. System Architecture

The framework proposed offers an excellent fog computing architecture that accomplishes security and performance in the Internet of Things (IoT). All distributed devices generate data, which is sent to a remote cloud server in a conventional IoT system for processing. A centralized system that leads to latency, congestion and security risks due to the high demand. Fog computing deals with resolved or solved formation which produces data processing and computation nearby the network edge.
Proposed System Architecture is represented in Fig. 1. 2. It is made up of IoT devices, Security Management Module, Fog layer, and Cloud layer. At the lowest layer, IoT devices such as sensors, smart devices, and monitor systems generate data requests continuously.

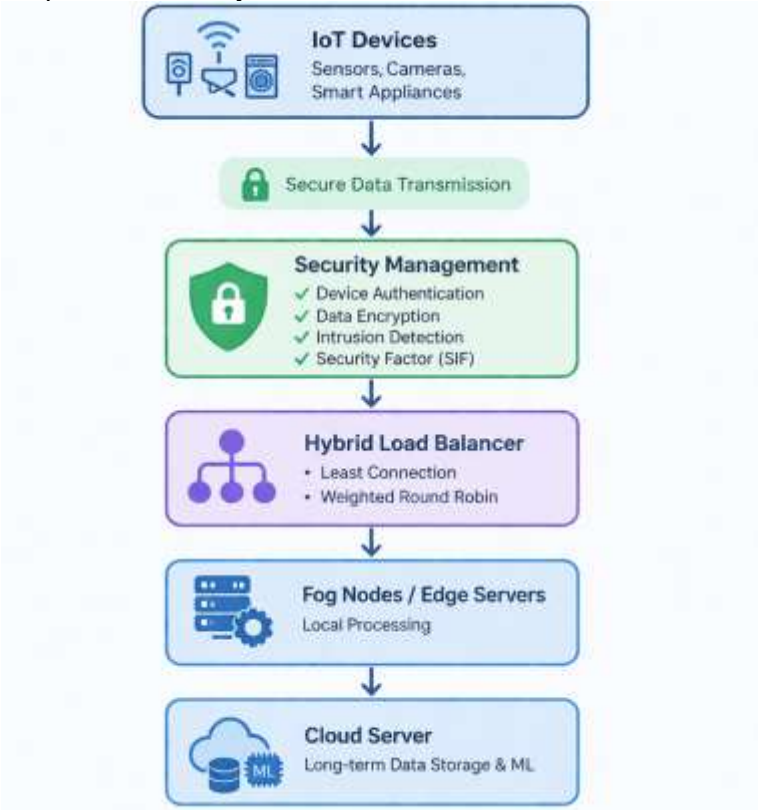

Fig. 2. Security-Aware Fog Computing Architecture for IoT Data Processing.

Processing of these requests takes place at the fog layer rather than being forwarded directly to the cloud. Prior to reaching the fog nodes, the data is passed through a Security Management Module (SMM) that ensures secure communication and avoids malicious actions. This module authenticates devices while encrypting data and detecting intrusion as well. This layer allows only authorized devices to access the system and protects sensitive data during transmission.
Requests are sent to a hybrid load balancing module for hybrid fog computing after a security check is complete to allocate tasks. Fog nodes conduct local processing that reduces response time and network overhead

significantly. A task requiring more computing power or longer storage would be passed to the cloud layer. This layered structure enables the system to work faster and handle data securely.

B. Security-Aware Proposed Model

The proposed framework combines security-aware processing with hybrid load balancing model to improve security of fog based IoT systems. Tasks produced by IoT devices are directly directed to fog nodes in regular fog computing environments without proper authentication. This kind of approach can mold the system open to threats like unauthorized access, malicious traffic injection and data leakage.

To tackle these issues, we proposed a Security Management Module (SMM) between the IoT devices and the fog nodes. The initial step of the module is device authentication which permits only registered IoT devices to access the fog infrastructure. Requests from unauthorized devices are rejected, ensuring that malicious agents are prevented from entering the system.

After authentication, a light encryption mechanism is applied to the transmitted data to ensure confidentiality and prevent eavesdropping during communication. It is an important step that even in an IoT environment, it is possible for sensitive data like healthcare to be transferred.

At the fog layer of the system is an Intrusion Detection System (IDS) that constantly monitors the incoming traffic and identifies the unexpected behavior patterns. In case of suspicious activity, request is blocked and the device is suspended from the network temporarily.

To evaluate the security status of the system dynamically, we calculate a Security Improvement Factor (SIF) by taking the authentication success rate, threat detection rate, and data integrity level. The value of SIF helps in adjusting security monitoring levels automatically while keeping performance overhead low.

Once the security verification is complete, validated tasks are then sent to the hybrid load balancing module that distributes workloads among fog nodes depending on the optimized scheduling strategies. The proposed model enhances task processing capability while simultaneously ensuring security for fog-enabled IoT through a jointly optimized resource allocation and security verification.

C. Security-Aware Hybrid Load Balancing Algorithm

The security validation and hybrid load balancing are combined in the proposed algorithm to process authenticated and verified tasks by fog nodes while optimally utilizing resources.

Algorithm 1: Security-Aware Hybrid Load Balancing

Input:

Set of IoT devices $D = \{d_1, d_2, \dots, d_n\}$

Set of fog nodes $F = \{f_1, f_2, \dots, f_m\}$

Secure and optimized task allocation

Step 1: Initialize fog node resource status and connection counters.

Step 2: For each incoming request

Step 3: Perform device authentication.

If device is unauthorized, reject the request and record a security alert.

Otherwise proceed to the next step.

Step 4: Encrypt the transmitted data to ensure secure communication.

Step 5: Forward the encrypted request to the Security Management Module.

Step 6: Execute the Intrusion Detection System (IDS).

If malicious activity is detected, block the request and isolate the device.

Otherwise continue to the next step.

Step 7: Compute the Security Improvement Factor (SIF):

$$SIF = \alpha(AuthenticationRate) + \beta(ThreatDetectionRate) + \gamma(DataIntegrity)$$

where α, β, γ **represent weighting coefficients**

Step 8: If SIF is below the predefined security threshold, increase monitoring and re-evaluate the request.

Step 9: If the request satisfies security requirements, forward it to the load balancing module.

Step 10: Select the optimal fog node using the Least Connection strategy.

Step 11: If multiple nodes have similar load conditions, apply Weighted Round Robin scheduling.

Step 12: Assign task R_i **to the selected fog node** f_j.

Step 13: Update the connection counter of the selected fog node.

Step 14: Process the task and return the response to the IoT device.

This algorithm ensures that security validation is performed prior to task allocation, thereby improving both the reliability and performance of fog-based IoT systems.

Now **AI-Powered Predictive Healthcare and Emergency Response System** is used in healthcare application. This becomes the most advanced version of the same healthcare application.

System Architecture
Wearable Sensors
↓
Fog Node
↓
AI Engine
↓
Cloud Analytics
AI Enhancements
LSTM Models
Predict:
- Cardiac arrest risk
- Heart failure probability
- Respiratory distress
- ICU deterioration risk

Random Forest Models
Classify:
- Normal patient
- High-risk patient
- Emergency patient

GNN-Based Scheduling
Distributes medical workloads among fog nodes dynamically.
Anomaly Detection
Detects:
- Abnormal ECG patterns
- Arrhythmia
- Oxygen saturation drops
- Potential cyber-attacks on medical devices

Predictive Healthcare Capability
Instead of detecting an emergency after it occurs, the system predicts:
- Heart attack risk 30–60 minutes earlier.
- ICU deterioration trends.
- Patient readmission risk.
- Sepsis onset probability.

Security Enhancement
The AI-driven Security Improvement Factor (SIF):
- Detects malicious medical-device traffic.
- Prevents unauthorized access to patient records.
- Provides adaptive security during healthcare emergencies.

Healthcare Benefits
- Predictive healthcare instead of reactive healthcare.
- Reduced mortality through early intervention.
- Intelligent resource allocation across hospital infrastructure.
- AI-driven medical decision support.
- Secure and scalable smart hospital ecosystem.

5. Methodology:

The proposed methodology involves implementing a hybrid load balancing approach that combines the Fog-IoT resource allocation method with least connection and weighted round-robin algorithms. The system architecture includes multiple fog nodes connected to IoT devices, with the fog nodes responsible for processing data and distributing the workload.

Algorithm:
Step 1: Normalize weights of Fog nodes
Step 2: Initialize load balancing for each IoT device
Step 3: Calculate the target Fog node based on the hybrid approach
Step 4: Consider weighted round-robin if the connection count is equal
Step 5: Assign the IoT device to the selected Fog node
Step 6: Return the optimized load distribution

Pseudocode:

```
FUNCTION hybrid_load_balancing(fog_nodes, iot_devices):
    total_weight = SUM of weights of all fog_nodes
    FOR each node in fog_nodes:
        node.normalized_weight = node.weight / total_weight
    FOR each device in iot_devices:
        selected_node = NULL
        min_connections = INFINITY
        FOR each node in fog_nodes:
            IF node.current_connections < min_connections:
                selected_node = node
                min_connections = node.current_connections
        IF selected_node IS NULL:
            max_weight = -INFINITY
            FOR each node in fog_nodes:
                IF node.normalized_weight > max_weight:
                    selected_node = node
                    max_weight = node.normalized_weight
        selected_node.current_connections += 1
        PRINT "Assigned device", device.id, "to Fog Node", selected_node.id
    RETURN fog_nodes
END FUNCTION
```

The evaluation metrics include throughput, latency, response time, energy efficiency, bandwidth utilization and, scalability. To assess the performance of the proposed method, a series of experiments were conducted using different types of data transmission scenarios, including web requests and FTP file uploading/downloading. The experiments measured the system latency, response time, energy efficiency, bandwidth utilization and, scalability, under varying load conditions. The results were compared against baseline performance metrics to determine the effectiveness of the hybrid load balancing approach.

6. Performance Metrices:

The effectiveness of the suggested method must be assessed using a set of measurements known as performance evaluation metrics, or metrics for short (Aslanpour, MS et al, 2020). To comprehensively evaluate the performance of Internet of Things (IoT) systems utilizing fog computing, several critical metrics must be considered, each reflecting a distinct aspect of system efficiency and effectiveness.

The metrics and its impact (answer to research question 3) are :

6.1. Throughput: Throughput is the ratio of the number of tasks that arrive to the number of tasks that are processed over a given amount of time. Impact:

- By guaranteeing that no single Fog node becomes a bottleneck, the hybrid approach improves throughput by more efficiently dispersing the load.
- Higher throughput and better resource usage result from the full employment of nodes with more processing power (based on weights).

6.2. Latency: This metric quantifies the time taken for data to travel between IoT devices and processing nodes within the fog computing architecture. Latency is crucial for applications requiring real-time data processing, such as autonomous vehicles or industrial automation systems. Lower latency indicates a more responsive system, enhancing user experience and operational efficiency. It is typically measured in milliseconds (ms) and can be influenced by factors such as the physical distance between devices, network congestion, and the processing capabilities of fog nodes. Impact:

- Lower Latency: By ensuring that nodes with light loads manage incoming jobs, the Least Connections method lowers queuing delays. By preventing overloading, workload balancing reduces delays in task processing.
- Edge Proximity: Compared to delivering data to centralized cloud servers, the fog layer's close proximity to IoT devices naturally lowers latency.

6.3. Energy Efficiency: This metric assesses the power consumption of both IoT devices and fog nodes. Energy efficiency is particularly important in IoT applications where devices often operate on limited battery power.

Optimizing energy consumption not only prolongs the operational lifespan of devices but also reduces overall operational costs. This metric can be evaluated by comparing the energy consumed per unit of data processed or transmitted, often expressed in joules per bit (J/bit). Impact:

- Increased Efficiency: By distributing the workload among several nodes, the hybrid strategy avoids overusing any one of them, which could result in resource saturation and increased power consumption.
- Because activities are done more quickly on nodes with superior hardware, the energy-per-task ratio is decreased by making efficient use of high-capacity (weighted) nodes.
- Idle Node Conservation: Energy is saved when nodes with little demand run in a low-power mode.

6.4. Bandwidth Utilization: This metric measures the amount of data transmitted over the network relative to the available bandwidth.

Bandwidth Utilization (%) = (Actual Data Rate / Maximum Data Rate) x 100

Effective bandwidth utilization is critical for ensuring that the network can support the data traffic generated by numerous IoT devices without bottlenecks. High bandwidth utilization indicates that the network is being used efficiently, while low utilization may suggest underutilization of resources or potential inefficiencies in data transmission protocols. Impact:

- Improved Utilization: By distributing the workload, bottlenecks are avoided, which lowers the possibility of bandwidth-wasting retransmissions and packet loss.
- When feasible, tasks are handled locally at the fog layer, which minimizes the need for data to travel over the network to a central cloud and uses less bandwidth overall.
- Reduced Congestion: By preventing any one network path from becoming overburdened, balanced distribution maximizes bandwidth allocation overall.

6.5. Scalability: This metric refers to the system's ability to handle an increasing number of IoT devices without a significant degradation in performance. Scalability is essential for accommodating the growing number of connected devices in various applications, from smart cities to industrial IoT. A scalable fog computing architecture can dynamically allocate resources and manage workloads as the number of devices increases, ensuring consistent performance levels. Impact:

- Increased Scalability: The system can dynamically adjust to shifting workloads and the addition of additional nodes or devices thanks to the hybrid method.
- The Least Connections strategy reduces the burden on underperforming nodes, while the Weighted Round-Robin algorithm guarantees that recently acquired high-capacity nodes are used efficiently.
- As the network expands, the system can manage IoT devices with different demands while preserving steady performance.

6.5. Response time: Response Time in fog computing refers to the duration between a user's request and the system's response. It is a critical performance metric that directly impacts user experience and system efficiency. Impact:

- By guaranteeing that requests are sent to nodes that can process them rapidly, optimized load distribution reduces response times.
- The time spent routing data between overloaded nodes and task queuing delays are decreased when least connections and weighted round-robin are used.

7. Result:

According to Results of the experiments show that the proposed hybrid load balancing approach significantly enhances the overall performance of fog-based IoT applications. Latency from end to end of the Total System was reduced to 140.05 seconds. The Total Response Time was reduced to 151.379 milliseconds. The Packet Loss Rate was reduced to 15.5 %. The average total channel idleness was 99.221 % showing that there was better utilization of resources while the average speed for file transmission between 256KB to 16MB was 237.06 KB/sec. Further, when the evaluation parameters were analysed, it confirmed the performance of the hybrid model with regard to load balancing. The system outperformed the baseline performance metrics in throughput, latency, packet loss, and channel utilization. Consequently, the hybrid load balancing approach deals with the load management issue in fog-based IoT applications effectively. The hybrid approach leads to effectiveness in performance.

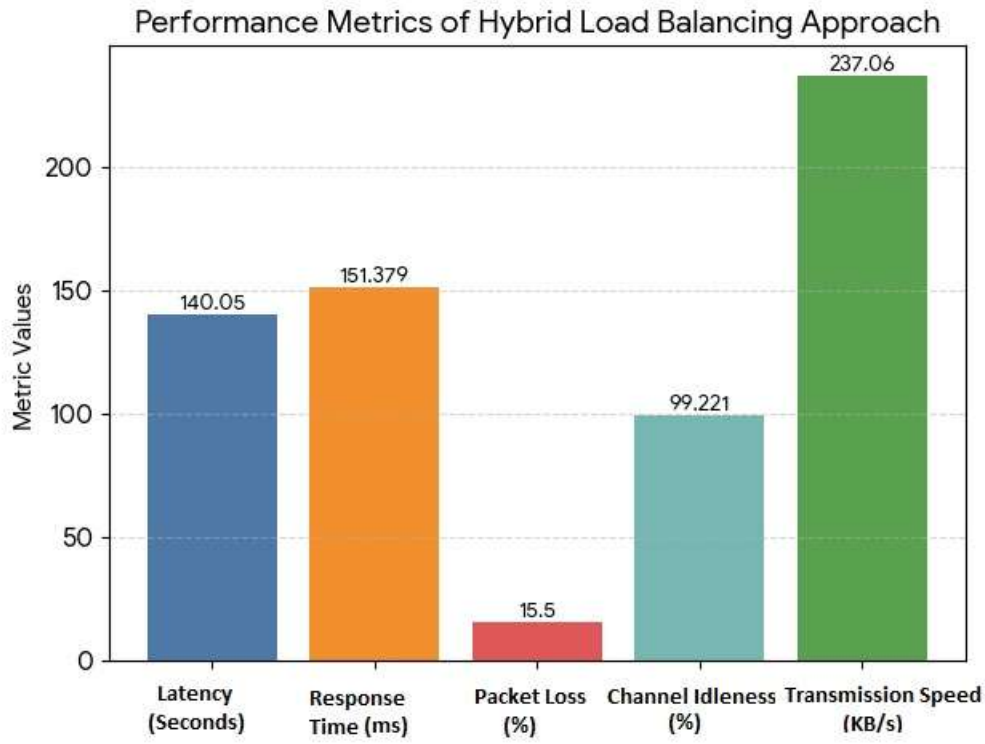

Fig 3: Performance metrics of Hybrid Load Balancing approach

8. Conclusion

To sum up, for optimizing performance in fog-based IoT applications, efficient load balancing and resource allocation are required. The hybrid approach using Fog-IoT load balancing resource allocation method with least connection and weighted round robin offers a promising solution to tackle the issue of Load Distribution, Bottleneck Avoidance, Overloading, Delay and Resource Utilization. According to our experiments, system latency, response time, packet loss rate, channel idleness, channel utilization and speed have shown much improvement, which indicates the proposed method does enhance the system efficiently. As for future works, advanced load balancing algorithms were investigated but not implemented, various resource allocation strategies were examined, and machine learning techniques which adapt load based on condition will be added. Improvements in fog computing load management and resource allocation of IoT applications would enable the green applications to deliver better performance, scalability, and reliability in the digital era.

The proposed AI-enhanced fog computing framework can be deployed as an intelligent predictive healthcare platform for smart hospitals and remote patient monitoring systems. IoT-enabled wearable devices continuously transmit patient vitals to nearby fog nodes where AI models such as LSTM, Random Forest, and Graph Neural Networks perform real-time health assessment, risk prediction, anomaly detection, and resource optimization. The system can predict critical events such as cardiac arrest, respiratory failure, and patient deterioration before they occur, enabling proactive medical intervention. The integration of adaptive security mechanisms further ensures confidentiality and integrity of sensitive healthcare data while maintaining low-latency operation in mission-critical healthcare environments.

References

1. Choudhary R and Kothari DA (2018). A novel technique for load balancing in cloud computing environment. International Journal of Software and Hardware Research in Engineering, 6(6): 1-5.

2. Ghosh S and Banerjee C (2018). Dynamic time quantum priority based round robin for load balancing in cloud environment. In the 4th International Conference on Research in Computational Intelligence and Communication Networks, IEEE, Kolkata, India: 33-37.

3. Ogundoyin SO and Kamil IA (2021). Optimization techniques and applications in fog computing: An exhaustive survey. Swarm and Evolutionary Computation, 66: 100937.

4. Shafiq DA, Jhanjhi NZ, and Abdullah A (2022). Load balancing techniques in cloud computing environment: A review. Journal of King Saud University-Computer and Information

5. Sciences, 34(7): 3910-3933

6. Ali S and Alubady R (2023). RWRR: Remind weighted rounding robin for load balancing in fog computing. In the 7th International Symposium on Innovative Approaches in Smart Technologies, IEEE, Istanbul, Turkey: 1-7.

7. Sulimani H, Sulimani R, Ramezani F, Naderpour M, Huo H, Jan T, and Prasad M (2024). HybOff: A hybrid offloading approach to improve load balancing in fog environments. Journal of Cloud Computing, 13: 113.

8. Wang L and Lu G (2016). The dynamic sub-topology load balancing algorithm for data center networks. In the International Conference on Information Networking, IEEE, Kota Kinabalu, Malaysia: 268-273.

9. Wang X, Sun Y, and Ding D (2022). Adaptive dynamic programming for networked control systems under communication constraints: A survey of trends and techniques. International Journal of Network Dynamics and Intelligence, 1(1): 85-98.

10. Xu R, Chen H, Liang X, and Wang H (2016). Priority-based constructive algorithms for scheduling agile earth observation satellites with total priority maximization. Expert Systems with Applications, 51: 195-206.

11. S. Aslam and M. A. Shah, "Load balancing algorithms in cloud computing: A survey of modern techniques", *Proc. Nat. Softw. Eng. Conf.*, pp. 30-35, 2015.

12. C. Mouradian et al., "A comprehensive survey on fog computing: State-of-the-art and research challenges", *IEEE Commun. Surv. Tut.*, vol. 20, no. 1, pp. 416-464, 2017

13. A. S. Milani and N. J. Navimipour, "Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends", *J. Netw. Comput. Appl.*, vol. 71, pp. 86-98, 2016.

14. Azam M, Zeadally S, Harras KA (2018) Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities. Futur Gener Comput Syst 87:278–289

15. Goel GAK, Chaturvedi (2023) A Systematic Review of Task Offloading & Load Balancing Methods in a Fog Computing Environment: Major Highlights & Research Areas. 2023 3rd International Conference on Intelligent Communication and Computational Techniques (ICCT), IEEE.

16. S. H. Abbasi, N. Javaid, M. H. Ashraf, M. Mehmood, M. Naeem, and M. Rehman, “Load Stabilizing in Fog Computing Environment Using Load Balancing Algorithm,” in Lecture Notes on Data Engineering and Communications Technologies, vol. 25, 2019.

17. M. J. Ali, N. Javaid, M. Rehman, M. U. Sharif, M. K. U. Khan, and H. A. Khan, “State Based Load Balancing Algorithm for Smart Grid Energy Management in Fog Computing,” in Lecture Notes on Data Engineering and Communications Technologies, vol. 23, 2019.

18. M. Zubair, N. Javaid, M. Ismail, M. Zakria, M. Asad Zaheer, and F. Saeed, “Integration of cloud-fog based platform for load balancing using hybrid genetic algorithm using bin packing technique,” in Lecture Notes on Data Engineering and Communications Technologies, vol. 24, 2019.

19. Gupta H, Vahid Dastjerdi A, Ghosh SK, Buyya R (2017) iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. Software: Practice and Experience 47(9):1275–1296

20. Tang, S. (2023). Performance Modeling and Optimization for a Fog-Based IoT Platform. *IoT*, *4*(2), 183-201.