# An advanced ensemble load balancing approach for fog computing applications

**Koppolu Ravi Kiran[1], Dasari Lokesh Sai Kumar[2], Veerapaneni Esther Jyothi[3], Nguyen Ha Huy Cuong[4], Nguyen Hoang Ha[5]**
[1]Department of Computer Science and Engineering, Jawaharlal Nehru Technological University, Andhra Pradesh, India
[2]Department of Computer Science and Engineering, P.V.P. Siddhartha Institute of Technology, Andhra Pradesh, India
[3]Department of Computer Applications, V.R. Siddhartha Engineering College, Andhra Pradesh, India
[4]Software Development Centre, The University of Da Nang, Da Nang, Viet Nam
[5]Development of Information Technology, Hue University of Sciences, Hue City, Viet Nam

**Article Info**

***Article history:***

Received Aug 8, 2023
Revised Nov 8, 2023
Accepted Dec 5, 2023

***Keywords:***

Adaptive load balancing
Cloud fog computing
Internet of things
Least-connections
Round-robin

**ABSTRACT**

Fog computing has emerged as a viable concept for expanding the capabilities of cloud computing to the periphery of the network allowing for efficient data processing and analysis from internet of things (IoT) devices. Load balancing is essential in fog computing because it ensures optimal resource utilization and performance among distributed fog nodes. This paper proposed an ensemble-based load-balancing approach for fog computing environments. An advanced ensemble load balancing approach (AELBA) uses real-time monitoring and analysis of fog node metrics, such as resource utilization, network congestion, and service response times, to facilitate effective load distribution. Based on the ensemble's collective decision-making, these metrics are fed into a centralized load-balancing controller, which dynamically adjusts the load distribution across fog nodes. Performance of the proposed ensemble load-balancing approach is evaluated and compared it to traditional load-balancing techniques in fog using extensive simulation experiments. The results demonstrate that our ensemble-based approach outperforms individual load-balancing algorithms regarding response time, resource utilization, and scalability. It adapts to dynamic fog environments, providing efficient load balancing even under varying workload conditions.



***Corresponding Author:***

Nguyen Ha Huy Cuong
Software Development Centre, The University of Da Nang
Da Nang, Viet Nam
Email: nhhcuong@sdc.udn.vn

## 1. INTRODUCTION

Fog computing, a paradigm facilitated by the cloud and geographically distributed infrastructure, facilitates the decentralization of networking capabilities and computational resources to the periphery of the network. This approach aims to bring these resources closer to end-users and internet of things (IoT) devices. [1]. Most cloud-only data processing, analysis, and storage architectures are sent to cloud servers, which may negatively impact latency, security, mobility, and dependability. Applications that are sensitive to location and time, the cloud struggles to satisfy its ultra-low latency needs. The fog layer's closeness to the IoT devices may significantly reduce latency [2], [3]. Fog computing constantly supports the cloud, forming new apps and services.

In fog computing settings, consumers demand fast-responding apps. We can employ fog networks to greatly enhance quality of service (QoS) because load balancing is efficient and can be considered vital. The purpose of fog load balancing is to [4] spread the load to fog nodes cloud, using one approach, to avoid fog node overload or underload. This technique maximizes throughput, performance, and resource use while minimizing responsiveness, expense, and energy use. Fog computing brings cloud computing to end-users. Cisco coined "fog" to describe real-life fog [5]. Fog computing employs the concept of situating the virtual fog platform in close proximity to end-users, therefore establishing an intermediary layer between their devices and the cloud. Clouds are situated at higher altitudes inside the Earth's atmosphere, while fog tends to occur in closer proximity to the Earth's surface. Fog computing facilitates the execution of computational tasks at the periphery of the network, hence enabling the transmission of novel services and applications that are specifically designed for the internet [6]. Studying fog computing architecture reference model is crucial. Fog computing designs, usually three-layered, have been proposed recently [7], [8]. In a fog network, a fog layer is positioned between cloud infrastructure and user devices, thereby extending cloud services to the network edge. Figure 1 depicts the framework of fog hierarchically [9], having three layers.

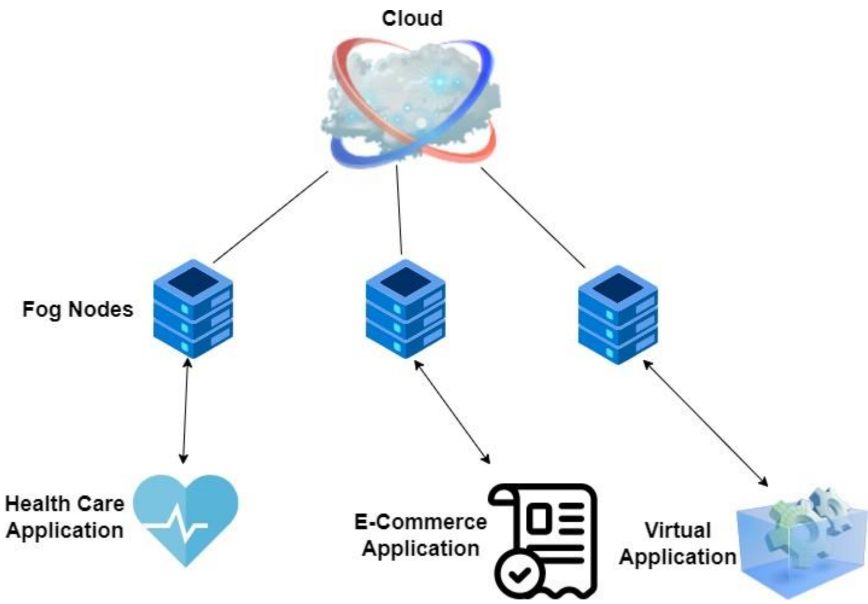


Figure 1. The framework of fog network

Fog computing has been recognized as a complementary technology to cloud computing, enabling the localization of processing and storage resources in greater proximity to the network's periphery. It aims to address the latency, bandwidth constraints, and network congestion issues that arise when processing data in the cloud. Fog computing employs a distributed architecture, with edge devices such as routers, gateways, and IoT devices performing computational tasks closer to the data source. In the context of fog computing, the execution of tasks is a significant aspect to consider the scheduling is critical for optimizing resource utilization and ensuring task execution efficiency. It entails assigning tasks to appropriate fog nodes based on processing capabilities, resource availability, network conditions, and application requirements. The goal is to reduce latency, increase throughput, and improve overall system performance. However, due to the dynamic nature of the fog environment, fog computing task scheduling is a complex problem. The availability and performance of fog nodes can fluctuate over time, as can the characteristics of tasks and their dependencies.

Several scheduling algorithms and techniques have been put forward to deal with the fog computing task scheduling problem. Heuristic-based algorithms, genetic algorithms, game theory-based approaches, and machine learning-based methods are among them. Each system has advantages and disadvantages, and the certain needs and constraints of the fog computing environment determine the scheduling algorithm chosen. In this paper, we will investigate the task scheduling problem in fog computing and review existing scheduling algorithms and techniques. We will examine their benefits, drawbacks, and applicability in various scenarios. In addition, we will discuss the challenges and future research directions in fog computing task scheduling to improve system performance and resource utilization even further. In the context of fog computing systems, the scheduling of tasks plays a crucial role in optimizing resource allocation and facilitating the provision of high-performance services with minimal latency. Fog computing has the capability to efficiently facilitate the operation of developing applications, including real-time analytics, edge artificial intelligence (AI), and IoT-based services, via the optimization of job allocation and scheduling choices. Task scheduling algorithms are critical in cloud computing because they efficiently allocate computational resources to execute tasks submitted by users or applications. Task scheduling is in charge of determining which tasks should complete, when they should complete, and what resources they should achieve. Task scheduling algorithms aim to maximize resource utilization, reduce task completion time, and ensure user or application fairness. Effective scheduling algorithms consider task characteristics, resource availability, and resource utilization, and user requirements. Cloud environments are typically made up of a diverse set of resources, such as virtual machines, containers, and physical servers. The high-performance computing-based task scheduling algorithms must efficiently manage these various resources. The contributions of the paper are listed: i) proposed an advanced ensemble load balanced approach (AELBA) for effective load balancing, ii) illustrated the AELBA combined with fog computing health care systems, and iii) analyzed the performance of AELBA.

The paper is structured as follows section 2 delineates many methodologies used in load balancing for fog computing. Section 3 described the working of AELBA approach and illustrated advantage of combining AELBA with fog computing health care systems. Section 4 focused on performance analysis of proposed approach, section 5 listed the conclusion of the study.

## 2. RELATED WORK

Mishra *et al.* [10] introduced the energy-based task allocation among multi-cloud networks that reduces the overall usage of energy. The energy usage is improved by using the proposed model up to 14.6%, 6.43%, and 2.18% to allocate the random approach such as cloud Z-score normalization (CZSN) and Fuzzy resource utilization (FR-MOS). Lilhore *et al.* [11] introduced the extended load balancing method that improves the response time, ready time and resource utilization. The proposed system utilized k-means clustering to cluster the virtual machines with less memory. Finally, the proposed model efficiently manages the virtual machines (VMs) and processes the user's requests. Kaur *et al.* [12] proposed an integration model that combines deep learning with other cloud domains that manage cloud load balancing. The optimized scheduling model creates the deep learning (DL) based method that schedules the task using the Genome workflow. The existing models focused on optimization techniques that utilize overflow VMs. It also reduces the latency, expenses, response time, and load balancing methods over machines. Nguyen *et al.* [13] proposed the workload estimated approach that predicts the work at the cloud VMs and how the load balancing approach manages the number of tasks at a time. The proposed model improved the prediction by adopting back-propagation to enhance efficiency in terms of projection. The proposed model also focused on reducing errors up to 0.001. Singh *et al.* [14] introduced a novel Quantum model that predicts the workload in the cloud data center. In this scenario, various issues are identified in computing the workload. To overcome this, a self-balanced adaptive differential evolution (SB-ADE) model was introduced to optimize the network weights. The accuracy is improved extensively compared with existing algorithms. The novel approach's performance was analyzed using eight datasets with different factors. The overall accuracy of the novel approach is about 91.45 percent which is high compared with existing systems. Cuong *et al.* [15] introduced the improved default weighted RR algorithm that executes multiple users by combining the JMeter to access the virtual server in the cloud domain. Finally, the proposed model has network load-balancing approaches with positive effects. Prakash *et al.* [16] submitted the dynamic multi-cloud system that runs the different cloud platforms. It is essential to complete the parallel tasks at a time. The goal is the proposed approach combined with distributed cloud with less time and low cost. Experiments show that the dynamic process gives accurate results compared with previous models.

Zahid *et al.* [17] proposed an architectural structure with three layers with distributed fog, centralized cloud, and consumer layers. Kamal *et al.* [18] introduced scheduling with little potential for conflicts in order to load-balance. This constraint satisfaction algorithm uses heuristics. Cloud, fog, and end- users form the architecture. Oueis *et al.* [19] employed fog load balancing to improve user experience. Multiple users need computation offloading; therefore, local computation cluster resources must handle all requests. Praveen *et al.* [20] recommended balancing burden on the cloud-fog infrastructure using VM scheduling. Praveen *et al.* [21] presented a fog-based PSO resource allocation approach for load balancing. Shi *et al.* [22] the use of fog and software-defined networking (SDN) architecture is proposed as a means to mitigate latency in cloud-based mobile face recognition systems. The authors also provided definitions for software-defined networking (SDN) and fog/cloud load balancing, characterizing them as optimization challenges. They proposed the use of a firework algorithm (FWA) that is based on SDN centralized control as a potential solution for this problem. In the context of a microgrid-connected wireless sensor network and fog environment, Karthik and Kavithamani [23] developed a whale optimization technique for load balancing. Krishna *et al.* [24] in this study proposed, a hybrid metaheuristic algorithm was used to effectively distribute the workload in fog-based vehicle ad hoc networks.

Li *et al.* [25] focused on examining the runtime elements of fog infrastructure and proposed the use of self-similarity-based load balancing (SSLB) as a recommended approach for managing the workload in large-scale fog systems. SSLB efficiency was achieved using an adaptive threshold policy and scheduling approach. Singh *et al.* [26] created a fuzzy logic load balancer with variable tuning settings and design hazy fog network controllers. Fuzzy logic model for link analysis as hubs for regulating movement. Abedin *et al.* [27] created a fog load balancing issue to reduce fog load balancing costs. Narrow band technology internet-connected devices (NB-IoT). The time resource scheduling issue in the Bankruptcy game NB-IoT. Subsequently, Beraldi *et al.* [28] presented a distributed a technique for balancing the burden of fog based on the random walk approach.

## 3. METHOD

Ensemble balancing workloads in fog computing environment involves distributing the workload across multiple nodes (devices or servers) in a fog network to optimize resource utilization, reduce latency, and improve overall system performance. Taking into account the dynamic and heterogeneous character of fog settings, sophisticated ensemble load balancing algorithms can be developed for fog computing. The AELBA algorithm typically operates in two stages: the task allocation phase and the load balancing phase. In the task allocation phase, tasks are assigned to the computing powers of the fog nodes and the nature of the tasks. This allocation can be performed using various techniques such as round-robin, least-loaded, or weighted fair queuing. After dividing up the work among the fog nodes, the load balancing phase comes into play. AELBA employs load balancing techniques to guarantee that the burden is evenly allocated among the fog nodes, optimizing their utilization as shown in Figure 2.

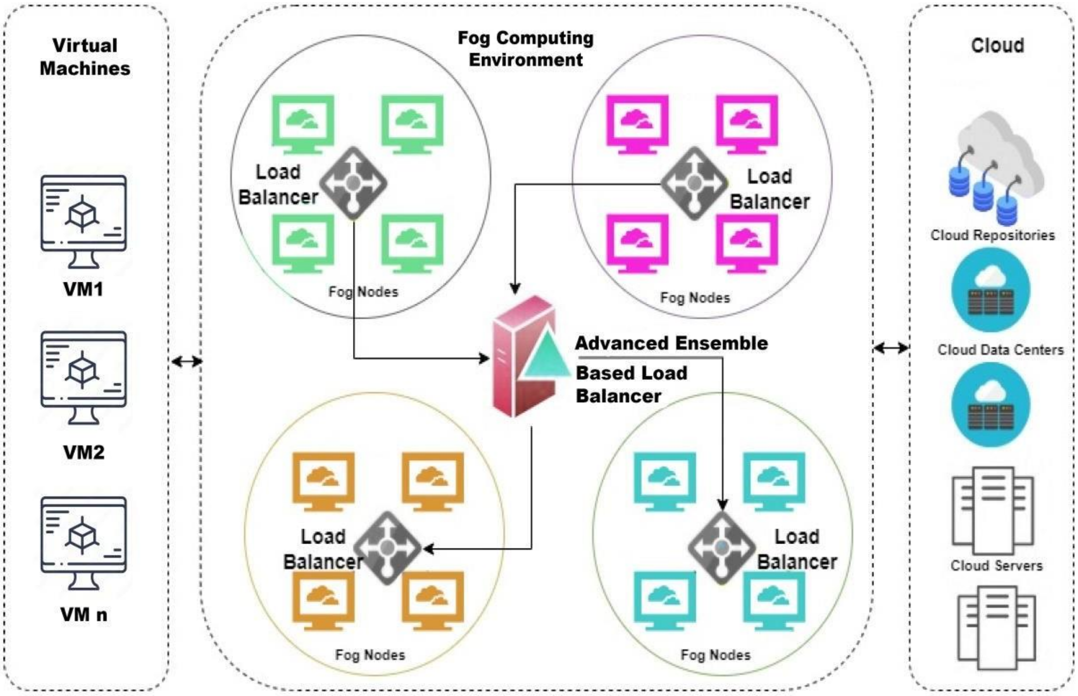


Figure 2. Proposed architecture for advanced ensemble based load balancer

The step by step procedure of AELBA is described:

Step 1: The approach begins by setting up the fog computing environment with multiple fog nodes distributed across a geographical area. Each fog node is equipped with load monitoring agents and communication capabilities. Load balancing parameters and thresholds are initialized, and communication channels between fog nodes for collaboration are established.

Step 2: Fog nodes continuously monitor their local resource utilization, such as central processing unit (CPU), memory, network bandwidth, and storage capacity. Additionally, they collect information about the overall system and network load. This real-time load monitoring provides an up-to-date view of the current workload and the supply of resources in the fog environment.
Step 3: Based on the current resource utilization of fog nodes and their performance capabilities, tasks or workloads are partitioned and assigned to appropriate fog nodes.
Step 4: Fog nodes communicate and collaborate with each other to exchange load and resource information.
Step 5: Unavailable or failed fog nodes are identified. If a fog node experiences a failure and becomes unavailable or disconnection, the system quickly redirects tasks to other available nodes, ensuring high availability and system resilience. Redundancy and failover strategies are employed to maintain uninterrupted service.
Step 6: If the workload exceeds the capacity of available fog nodes, dynamic resource provisioning is triggered. This involves the on-demand allocation of additional resources, such as virtual machines or containers, to handle the increased workload efficiently.
Step 7: The entire load balancing process is performed iteratively and continuously. As the workload and network conditions change, the load balancing approach the system constantly adapts the allocation of tasks in order to optimize the use of resources and minimize response times.

### 3.1. AELBA combined with fog computing health care systems

Healthcare systems play a critical role in providing quality care to patients. With the advancements in technology, people have become more interested in incorporating fog computing and advanced load balancing approaches to improve the performance and efficiency of healthcare systems. This article introduces the concept of combining AELBA with fog computing in healthcare systems.

#### 3.1.1. Fog computing in healthcare systems

Fog computing is a decentralized computing environment that enables the deployment of processing resources, storage, and networking in closer proximity to the perimeter of the network, which is closer to the site where data is produced. Fog computing is also known as fog computing. For the purpose of processing and analyzing real-time patient data, fog computing may be used in the context of healthcare systems. This enables quicker reaction times, lower latency, and enhanced dependability.

#### 3.1.2. Combining AELBA with fog computing

By integrating the adaptive ensemble learning-based algorithm (AELBA) with fog computing, healthcare systems can unlock a myriad of substantial advantages. AELBA, leveraging its capacity to amalgamate diverse learning algorithms, provides an intelligent and adaptable framework for processing healthcare data. When synergistically employed with fog computing, which extends computational capabilities to the edge of the network, these technologies create a powerful alliance that addresses key challenges and offers transformative benefits for the healthcare sector. Here are some key advantages:

a) Improved performance: AELBA intelligently distributes tasks among fog nodes based on factors such as node capacity, workload, and network conditions. This dynamic load balancing improves overall system performance, reducing response times and enhancing user experience.
b) Enhanced reliability: Fog computing ensures that critical healthcare services can be provided even in the presence of network disruptions or failures. AELBA further enhances reliability by redistributing tasks to other available nodes if a particular node becomes unavailable or experiences performance issues.
c) Real-time data processing: Fog computing facilitates the processing of data in real-time at the network edge, minimizing the necessity for transmitting sensitive patient data to centralized servers. AELBA ensures that processing tasks are allocated to the most suitable fog node, minimizing latency and enabling faster decision-making.
d) Scalability: Healthcare systems often experience varying workloads and demands. By leveraging fog computing and AELBA, the system can dynamically scale resources up or down, adapting to changing requirements without compromising performance.

The combined approach of AELBA and fog computing can be applied to various healthcare scenarios, such as remote patient monitoring, telemedicine, and wearable healthcare devices. For example, patient data collected by wearable devices can be processed in real-time by fog nodes using AELBA, enabling continuous monitoring and timely interventions. The integration of AELBA with fog computing in healthcare systems offers significant advantages in terms of improved performance, reliability, real-time data processing, and scalability. This combined approach has the potential to enhance healthcare services, enable faster decision-making, and ultimately improve patient outcomes. As technology continues to advance, the adoption of such innovative solutions becomes increasingly important for the future of healthcare systems.

## 4. RESULT AND DISCUSSION

The implementation of cloud health care systems is developed by using Python and CloudSim simulators. AELBA load balancing mechanism is contrasted with popular load balancing algorithms round-robin (RR) and advanced task allocation scheduler (ATAS). RR algorithm functions by keeping a list or queue of available servers. When a new request comes in, it is routed to the next server on the list. The algorithm then updates the list, pushing the assigned server to the end of the queue and directing the following request to the next server. This cyclical process is repeated until each server receives an equal number of requests. The limitations of RR are that regardless of their capabilities, all servers are treated equally and receive an equal share of incoming tasks or requests in a cyclic manner.

Advanced task allocation scheduler (ATAS) is an intelligent scheduling mechanism that dynamically allocates tasks in a cloud computing environment. It employs a variety of algorithms and techniques to optimize task assignment to available resources, ensuring efficient resource utilization and meeting application requirements. ATAS's primary goal is load balancing, which entails distributing tasks evenly across available resources to avoid overloading or underutilization. It makes informed decisions about task allocation by considering resource availability, task characteristics, and application priorities. ATAS incorporates adaptive and learning capabilities, allowing it to respond quickly to changing workload patterns and resource availability. It constantly monitors system performance and modifies task allocation strategies as needed. ATAS can optimize resource allocation and overall system performance by dynamically adapting to workload variations. The limitation of ATAS is that increased complexity and computational overhead, which may impact its real-time responsiveness and efficiency in highly dynamic environments.

### 4.1. Performance metrics

In this section, the performance metrics for evaluating and optimizing the efficiency and effectiveness of cloud-based services. These metrics help measure various aspects of cloud performance to ensure that the system meets the desired objectives and delivers a satisfactory user experience. Here are some common performance metrics used in cloud computing:

a. Response time (RT): It measures the time taken for a request to be processed and responded to by the cloud infrastructure. Lower response times generally indicate better performance.

$$RT = Time\ of\ completion - Time\ of\ submission$$

b. Latency (LT): It refers to the delay between the initiation of a request and the start of a response. Lower latency indicates faster communication and better user experience.

$$LT = Time\ of\ arrival - Time\ of\ departure$$

c. Throughput (TP): This metric measures the amount of data or requests that can be processed within a given time frame. Higher throughput indicates better performance and scalability.

$$TP = Number\ of\ tasks\ completed / Time\ taken$$

Various tools and techniques, such as performance monitoring software, log analysis, and performance testing methodologies, can monitor, analyze, and optimize these performance metrics. By tracking and optimizing these metrics, cloud service providers and users can ensure that the cloud infrastructure performs optimally and meets the desired service-level objectives.

### 4.2. Experimental results

Table 1 depicts the performance of AELBA in comparison with RR and ATAS based on the performance metrics response time, latency and throughput. According to experimental results AELBA takes less time in comparison to RR and ATAS. Figures 3, 4, and 5 depicts the results of comparison.

Table 1. Performance comparison of AELBA with RR and ATAS

| Algorithms | RR | ATAS | AELBA |
|---|---|---|---|
| RT | 23.23 | 19.23 | 17.12 |
| LT | 27.67 | 25.12 | 23.23 |
| TP | 34.56 | 45.12 | 57.67 |

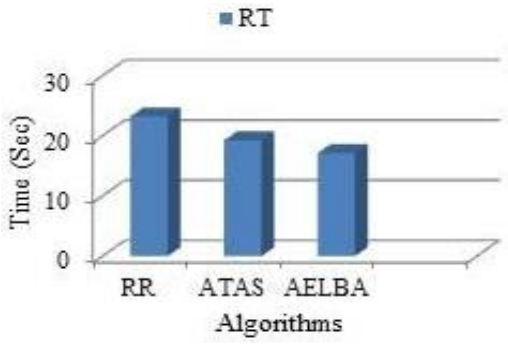


Figure 3. Comparison between existing and proposed models based on RT (seconds)

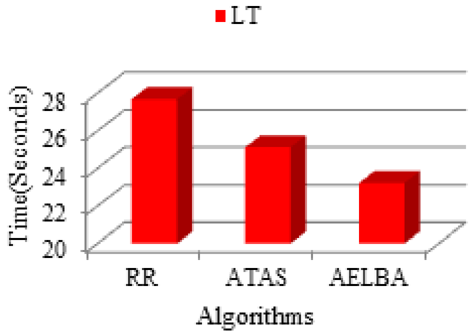


Figure 4. Comparison between existing and proposed models based on LT (seconds)

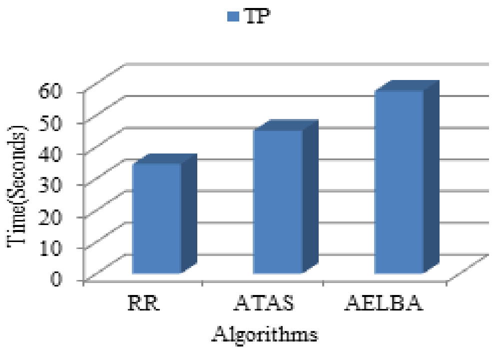


Figure 5. Comparison between existing and proposed models based on throughput (seconds)

## 5. CONCLUSION

This paper presents an ensemble cloud load-balancing framework designed for fog computing environments. By leveraging the collective intelligence of multiple load-balancing algorithms, our approach ensures optimized resource utilization and improved performance in fog computing systems. The proposed ensemble load balancing solution contributes to the advancement of fog computing technologies, making them more scalable, reliable, and efficient for a diverse array of internet of things applications. AELBA is a very efficient and complex approach for balancing workloads in the cloud. It utilizes ensemble learning to achieve best use of resources and division of work across multiple servers. AELBA incorporates a various load-balancing methods and techniques, including dynamic threshold-based load balancing, round-robin scheduling, and intelligent workload prediction. By leveraging these approaches together, AELBA can provide efficient utilization of resources, improved performance, and enhanced scalability in cloud environments. One of the critical advantages of AELBA is its capacity to adjust to different circumstances and workload patterns and adjust the load-balancing strategy accordingly. This adaptability ensures that resources are efficiently allocated based on real-time demands, preventing any individual server's overloading and minimizing user response time. Furthermore, AELBA incorporates intelligent workload prediction, which utilizes historical data and machine learning algorithms to forecast future resource demands accurately. AELBA optimizes resource utilization and prevents potential bottlenecks by proactively allocating resources based on these predictions. Overall, AELBA offers a comprehensive and advanced solution for balancing the loads in cloud environment. Its ensemble-based approach, dynamic adaptation, andintelligent workload prediction make it a powerful tool for achieving high performance, scalability, and smart use of resources in cloud environments. Implementing AELBA can significantly improve the overall performance of the system as well as improving the experience of using cloud-based apps for users.

## REFERENCES

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, Aug. 2012, pp. 13–16, doi: 10.1145/2342509.2342513.

[2] W. Saber, W. Moussa, A. M. Ghuniem, and R. Rizk, "Hybrid load balance based on genetic algorithm in cloud environment," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 3, pp. 2477–2489, 2021, doi: 10.11591/ijece.v11i3.pp2477-2489.

[3] K. Fizza, N. Auluck, and A. Azim, "Improving the schedulability of real-time tasks using fog computing," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 372–385, Jan. 2022, doi: 10.1109/TSC.2019.2944360.

[4] S. Aslam and M. A. Shah, "Load balancing algorithms in cloud computing: A survey of modern techniques," in *2015 National Software Engineering Conference (NSEC)*, Dec. 2015, pp. 30–35, doi: 10.1109/NSEC.2015.7396341.

[5] M. Rahimi, M. Songhorabadi, and M. H. Kashani, "Fog-based smart homes: A systematic review," *Journal of Network and Computer Applications*, vol. 153, Mar. 2020, doi: 10.1016/j.jnca.2020.102531.

[6] A. M. Alsmadi *et al.*, "Fog computing scheduling algorithm for smart city," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 3, pp. 2219–2228, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2219-2228.

[7] P. Kanani and M. Padole, "Exploring and optimizing the fog computing in different dimensions," *Procedia Computer Science*, vol. 171, pp. 2694–2703, 2020, doi: 10.1016/j.procs.2020.04.292.

[8] P. Sherubha *et al.*, "An efficient unsupervised learning approach for detecting anomaly in cloud," *Computer Systems Science and Engineering*, vol. 45, no. 1, pp. 149–166, 2023, doi: 10.32604/csse.2023.024424.

[9] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, 2018, doi: 10.1109/COMST.2017.2771153.

[10] S. K. Mishra *et al.*, "Energy-aware task allocation for multi-cloud networks," *IEEE Access*, vol. 8, pp. 178825–178834, 2020, doi: 10.1109/ACCESS.2020.3026875.

[11] U. K. Lilhore, S. Simaiya, K. Guleria, and D. Prasad, "An efficient load balancing method by using machine learning- based VM distribution and dynamic resource mapping," *Journal of Computational and Theoretical Nanoscience*, vol. 17, no. 6, pp. 2545–2551, Jun. 2020, doi: 10.1166/jctn.2020.8928.

[12] A. Kaur, B. Kaur, P. Singh, M. S. Devgan, and H. K. Toor, "Load balancing optimization based on deep learning approach in cloud environment," *International Journal of Information Technology and Computer Science*, vol. 12, no. 3, pp. 8–18, Jun. 2020, doi: 10.5815/ijitcs.2020.03.02.

[13] H. H. C. Nguyen, B. T. Khiet, V. L. Nguyen, and T. T. Nguyen, "An effective method for clustering-based web service recommendation," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, pp. 1571–1578, Apr. 2022, doi: 10.11591/ijece.v12i2.pp1571-1578.

[14] A. K. Singh, D. Saxena, J. Kumar, and V. Gupta, "A quantum approach towards the adaptive prediction of cloud workloads," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 12, pp. 2893–2905, Dec. 2021, doi: 10.1109/TPDS.2021.3079341.

[15] H. H. C. Nguyen, H. V. Dang, N. M. N. Pham, V. S. Le, and T. T. Nguyen, "Deadlock detection for resource allocation in heterogeneous distributed platforms," in *Recent Advances in Information and Communication Technology 2015: Proceedings of the 11th International Conference on Computing and Information Technology (IC2IT)*, 2015, pp. 285–295, doi: 10.1007/978-3-319-19024-2_29.

[16] P. Prakash, K. G. Darshaun, P. Yaazhlene, M. V Ganesh, and B. Vasudha, "Fog computing: Issues, challenges and future directions," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 6, pp. 3669–3673, Dec. 2017, doi: 10.11591/ijece.v7i6.pp3669-3673.

[17] M. Zahid, N. Javaid, K. Ansar, K. Hassan, M. K. Khan, and M. Waqas, "Hill climbing load balancing algorithm on fog computing," in *Advances on P2P, Parallel, Grid, Cloud and Internet Computing: Proceedings of the 13th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2018)*, 2019, pp. 238–251, doi: 10.1007/978-3-030-02607-3_22.

[18] M. A. S. Donepudi, V. Shariff, V. K. Pratap, S. Phani, and N. Praveen, "Security model for cloud services based on a quantitative governance modelling approach," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 7, 2023.

[19] S. P. Praveen, H. Ghasempoor, N. Shahabi, and F. Izanloo, "A hybrid gravitational emulation local search-based algorithm for task scheduling in cloud computing," *Mathematical Problems in Engineering*, vol. 2023, pp. 1–9, Feb. 2023, doi: 10.1155/2023/6516482.

[20] S. P. Praveen, B. Thati, C. Anuradha, S. Sindhura, M. Altaee, and M. A. Jalil, "A novel approach for enhance fusion based healthcare system in cloud computing," *Journal of Intelligent Systems and Internet of Things*, vol. 12, no. 1, pp. 88–100, 2023, doi: 10.54216/JISIoT.090106.

[21] D. Baburao, T. Pavankumar, and C. S. R. Prabhu, "Load balancing in the fog nodes using particle swarm optimization-based enhanced dynamic resource allocation method," *Applied Nanoscience*, vol. 13, no. 2, pp. 1045–1054, Feb. 2023, doi: 10.1007/s13204-021-01970-w.

[22] C. Shi, Z. Ren, and X. He, "Research on load balancing for software defined cloud-fog network in real-time mobile face recognition," in *Communications and Networking: 11th EAI International Conference, ChinaCom 2016 Chongqing, China, September 24–26, 2016, Proceedings, Part II 11*, 2018, pp. 121–131, doi: 10.1007/978-3-319-66628-0_12.

[23] S. S. Karthik and A. Kavithamani, "Fog computing-based deep learning model for optimization of microgrid-connected WSN with load balancing," *Wireless Networks*, vol. 27, no. 4, pp. 2719–2727, May 2021, doi: 10.1007/s11276-021-02613-2.

[24] T. B. M. Krishna, S. P. Praveen, S. Ahmed, and P. N. Srinivasu, "Software-driven secure framework for mobile healthcare applications in IoMT," *Intelligent Decision Technologies*, vol. 17, no. 2, pp. 377–393, May 2023, doi: 10.3233/IDT-220132.

[25] C. Li, H. Zhuang, Q. Wang, and X. Zhou, "SSLB: Self-similarity-based load balancing for large-Scale fog computing," *Arabian Journal for Science and Engineering*, vol. 43, no. 12, pp. 7487–7498, Dec. 2018, doi: 10.1007/s13369-018-3169-3.

[26] S. P. Singh, A. Sharma, and R. Kumar, "Design and exploration of load balancers for fog computing using fuzzy logic," *Simulation Modelling Practice and Theory*, vol. 101, May 2020, doi: 10.1016/j.simpat.2019.102017.

[27] S. F. Abedin, A. K. Bairagi, M. S. Munir, N. H. Tran, and C. S. Hong, "Fog load balancing for massive machine type communications: A game and transport theoretic approach," *IEEE Access*, vol. 7, pp. 4204–4218, 2019, doi: 10.1109/ACCESS.2018.2888869.

[28] R. Beraldi, C. Canali, R. Lancellotti, and G. Proietti Mattia, "Randomized load balancing under loosely correlated State information in fog computing," in *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Nov. 2020, pp. 123–127, doi: 10.1145/3416010.3423244.

**BIOGRAPHIES OF AUTHORS**

**Koppolu Ravi Kiran** is working as assistant professor in Department of CSE, Jawaharlal NehruTechnological University, Kakinada, Andhra Pradesh, India. He obtained his M.Tech in CSE from the JNTUK, Kakinada, India. He is currently pursing Ph.D. from JNTUK, Kakinada. He obtained his M.Tech in CSE from JNTUK, Kakinada, INDIA. He can be contacted at email: kravi1189@gmail.com.

**Dasari Lokesh Sai Kumar** is working as assistant professor in Department of CSE,P.V.P. Siddhartha Institute of Technology, Kanuru, Vijayawada. He is currently pursuing Ph.D from JNTUK, Kakinada. He obtained his M.Tech in CSE from JNTUK, Kakinada, INDIA. He is having 12 years of teaching experience. His research interests include, machine learning and deep learning. He can be contacted at email: lokeshsaikumar@gmail.com.

**Veerapaneni Esther Jyothi** was awarded Ph.D. from Rayalaseema University and M.Tech from JNTUK, Kakinada. She has 14 years of teaching experience. At present, she is working as assistant professor, Department of Computer Applications, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada. Her area of interest includes machine learning and network security. She can be contacted at email: vejyothi@vrsiddhartha.ac.in.

**Nguyen Ha Huy Cuong** obtained his doctorate in Computer Science/Resource Allocation Cloud Computing in 2017 from the University of Danang. He has published over 50 research papers. His main research interests include the resource allocation, detection, prevention, and avoidance of cloud computing and distributed systems. Currently, he is working at Software Development Centre, The University of Danang. He can be contacted at email: nhhcuong@vku.udn.vn.

**Nguyen Hoang Ha** was born in 1976 in Vietnam. Working in the Department of Information Technology, Hue University of Sciences. 1995-1999 Bachelor of Science (B.S) in science (majoring in computer science), Hue University of Sciences. 2003-2005 Master of Science (M.S) in computer science, Hue University of Sciences. 2012-2017 Doctor of Science (D.S) in computer science, Hue University of Sciences. His main research interests include Cloud Computing, parallel processing, and distributed processing. She can be contacted at email: nhha@husc.edu.vn.