Journal of Information Systems and Telecommunication
Vol.12, No.4, October-December 2024, 264-279
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Load Balancing Algorithms in Cloud, Fog Computing and Convergence of Fog and Cloud – A Survey

Seyedeh Leili Mirtaheri[1*], Mahya Azari Jafari [2] , Sergio Greco [1], Ehsan Arianyan[3], Reza Mansouri[4]

[1]. Department of Computer Engineering, Modeling, Electronics and Systems, University of Calabria, Italy
[2]. Faculty of Mathematics and Computer Science, Kharazmi University, Tehran, Iran
[3]. Department of Information Technology, ICT Research Institute (ITRC), Tehran, Iran
[4]. Computer Science Department, Georgia State University, Atlanta, GA, USA

Received: 04 Aug 2022/ Revised: 04 Oct 2023/ Accepted: 18 Nov 2023

Abstract
Cloud computing and fog computing are deployed as computing storage and services for the end-users. Fog computing promotes task performance through storage, computing, and networking services. Instead of taking place in centralized cloud computing data centers, these services can be provided via near-edge devices. Efficient load balancing in distributed computing systems has been the main challenge. The load balancing algorithm has an important role in enhancing the Quality of Service (QoS), throughput, and resource utilization and diminishing the potential cost and its strategy and architecture completely depend on the centralized or distributed architecture of the system and the type of requests. Cloud computing and fog computing use centralized and distributed architectures, respectively. The load balancing algorithm in these two environments cannot be the same. Meanwhile, the demand for near real-time processing requests is drastically increasing; load balancing should be able to handle real-time requests. This paper reviews and investigates the modern and diverse load balancing aspects of fog and cloud computing systems. We also categorize the load balancing algorithms in cloud and fog computing: meta-heuristic algorithms, heuristic algorithms, learning algorithms, and customized algorithms. We propose different research classes about the algorithm's type, objectives, simulation tools, and so forth. This review demonstrates that the most prevalent categories of methods used in load balancing in fog and cloud computing are custom approaches and meta-heuristic algorithms, respectively. While the most renowned load balancing algorithms have not yet succeeded in fog environments, meta-heuristic algorithms have shown their competence in cloud environments impeccably.

Keywords: Fog Computing; Cloud Computing; Convergence of Fog and Cloud; Load Balancing.

1- Introduction

Today, the dramatic development of IoT and mobile internet has caused both objects and people to connect to the internet anytime, anywhere. The substantial number of devices connected to the internet has led to tremendous data. Due to this vast amount of data, current processing and storage equipment cannot meet people's demands, making it difficult to manage them with current technology, including distributed systems and cloud computing.
Cloud computing is a suitable option for data processing because of its high storage and processing potential. Nonetheless, this processing pattern is centralized, and all processing of tasks must be performed literally in a cloud. It means that all requests are sent to a centralized cloud. The centralized point is a challenging issue in cloud computing because processing resources are not proportional to the network bandwidth [1].
In some applications of IoT, intelligent traffic control systems, smart homes, health-related systems, smart networks, and many other delayed-sensitive systems, we require low latency and mobility. Therefore, the delay is not acceptable to the system caused by exchanging the data with a centralized cloud [2].
Some cloud decisions can be calculated and implemented locally without being transmitted to the cloud, and the near real-time decision-making process cannot tolerate delay. Thus, Fog computing is a promising solution to support: 1) computational demand in real-time and sensitive applications, 2) delays in IoT and geographically distributed devices, 3) high-density network challenges, 4) long service delays, and 5) reduced quality of service [1].
Fog computing is a distributed computational model. This computational model places many heterogeneous network-

✉ **Corresponding Author**
Email Address

connected devices at the network's edges to provide services such as processing, network communication, and storage in a comprehensive manner. Thus, fog computing improves the system's overall performance. Fog computing responds effectively to near real-time applications and improves latency and bandwidth.

To assist future load balancing researchers in fog computing, we surveyed convergence fog, cloud computing, the various infrastructures, mechanisms, and existing algorithms in load balancing. This paper provides a new classification of load balancing algorithms in cloud and fog environments.

In sections 2 and 3 we go through fog and cloud computing definitions with various infrastructures, platforms, and technical aspects. In section 4, diverse load balancing techniques, their advantages, and load balancing metrics are presented. Section 5 is devoted to various classifications of load balancing algorithms. Section 6 gives different analyses of the research done, based on different categories, and finally, section 7 is the conclusion.

2- Cloud Computing

This section studies computational infrastructure and platforming aspects of cloud computing.

2-1- Definition

Cloud computing, as described by the National Institute of Standards and Technology (NIST), is a technology model which facilitates "convenient, resource pooling, ubiquitous, on-demand access which can be easily delivered with different types of service provider interaction."

2-2- Cloud Computing Infrastructure

Public cloud: Public cloud gives open and unrestricted access to infrastructure to the public [2],[3]. Private cloud: When computing takes place inside the data center, it is a private cloud. Community cloud: This model allows the cloud resources to be shared and utilized by more than one organization simultaneously. Virtual private cloud: It is a semi-private cloud deployment model with less infrastructure. Hybrid cloud: It is a combination of two or more clouds (public, private, or community) [3].

2-3- Service Models in Cloud Computing

IaaS (Infrastructure as a Service): [4]
PaaS (Platform as a Services): [4]
SaaS (Software as a Service): [4]
CaaS (Computing as a Service): [5]
SECaaS (Security as a Service): [5]

3- Fog Computing

This section studies computational infrastructure and platforms, features, and architecture of fog computing.

3-1- Definition

Fog computing is a model with constraints on storage, computing, and distributed network services between different devices and classic cloud computing [6]. The OpenFog Consortium elucidates fog computing as a system-level horizontal architecture, distributing storage, computing, control, and networking resources and services along the spectrum from cloud to things.

3-2- Fog Computing Infrastructure

According to the definition in [7], fog infrastructure has four types: private fog, public fog, community fog, and hybrid fog.

Private fog: Created and owned by an organization, a third party, or both, a private fog is deployable off or on-premises. While the fog is managed and operated by its owner, a single organization offers the resources exclusively (e.g., business units).

- Public fog: Created and owned by a government organization, company, academic institute, or a mixture, a public fog is deployed on the properties of the providers. While the fog is managed and operated by its owner, the general public offers the resources for open use.
- Community fog: Created by one or many organizations in a community, a third party, or an amalgamation of them, a community fog may be deployed off or on-premises. While the fog is maintained and operated by its creator(s), the resources are offered exclusively to consumers of a particular community of organizations with shared incentives.
- Hybrid fog: A type of fog computing that integrates a private/public cloud (i.e., a hybrid cloud) with a private/public/community fog, that can be proper due to the physical resource restraints. Consequently, this platform is extended in a scalable architecture as a hybrid cloud, that is elastic, scalable, and with available on-demand resources [7].

3-3- Service Models

Depending on who provides infrastructure, platform, or software, fog computing platforms can be classified as high/low-level virtualized resources in three different categories [7]. We classify fog computing workloads into static and dynamic ones—the last of which contain some metrics like the user, location, and time. In Figure 1, fog computing's service model is presented.

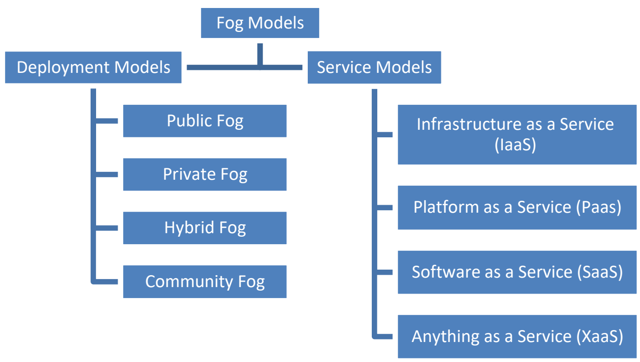

Fig. 1 Service model in fog

3-4- Characteristics of Fog Computing

According to the definitions of fog-based computing, a prominent aspect of this computational model is the proximity of resources to end devices (sensors/Internet of Things devices), which is one of the highlights compared to other computational models. Other fog computing aspects include real-time interactions, low latency, mobility, interoperability, scalability, geographical distribution, heterogeneity, security, low bandwidth consumption, and low energy consumption [6].

Fog-based processing is a new paradigm that tries to expand cloud computing capabilities at the network's edges. Performing a task via cloud computing may take a long time, especially when the network delay is high or the client's load is exceeded.

This case is more sensible in mobile devices because the wireless network delay is higher due to the relatively lower bandwidth. Therefore, researchers advanced the fog computing pattern to solve the problems regarding mobile devices. This computational model can improve performance while reducing energy consumption in environments where mobile devices are available [1]. There are some fog-based hierarchical architectures that add a layer of fog in the middle of cloud centers and end devices. Figure 2 shows the fog-based hierarchical architecture.

4- Load Balancing

Currently, load balance is a significant challenge in cloud computing. There are many requests from thousands of users and customers that need a lot of hardware and bandwidth. A load balancer helps to allocate the workload between different nodes and guarantee that no nodes are overloaded. A load-balancing algorithm's aim is improving the response time by using available resources. Other goals of load balancing algorithms are reducing computational time, increasing throughput, reducing error tolerance and execution time, and so on.

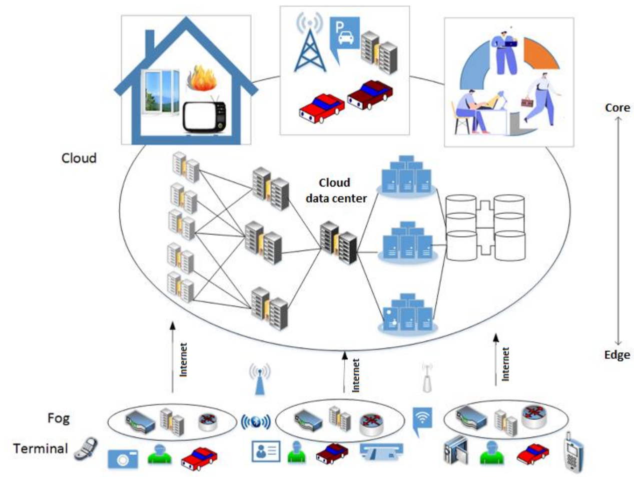

Fig. 2 Fog-based hierarchical architecture

4-1- Advantages of Load Balancing

Load balancing the system's workload improves all computational nodes' efficiency, thus improving its overall efficiency. Some significant advantages of load balancing are as follows [7], [8]:

• The task of waiting time is reduced.
• The task of response time is minimized.
• The exploitability of system resources is maximized.
• The system throughput is maximized.
• The readability and stability of the system are improved.
• It accommodates future modifications.
• Prolonged starvation is avoided for small jobs.
• In load balancing, overall system performance is enhanced by improving the performance of each node.

4-2- Load Balancing Metrics

Some important load balancing metrics are throughput, response time, scalability, resource utilization, fault tolerance, migration time, performance, overload, and energy consumption.

4-3- Types of Load Balancing Algorithms

Contingent upon the initiation of the process, load balancing algorithms are categorizable as follows:

• Sender-initiated: In this type, the sender initiates the process. The sender sends request messages until it finds a receiver accepting the load [9].
• Receiver-initiated: In this type of description algorithm, the process is initiated by the receiver, where it sends request messages until a sender able to get the load is found (a node that is under-loaded) [9].
• Symmetric: This type is an amalgamation of sender-initiated and receiver-initiated algorithms [9].

Subject to the system's current state, load balance algorithms may be categorized into dynamic and static as well [10]. Figure 3 depicts the taxonomy of load balancing algorithms.

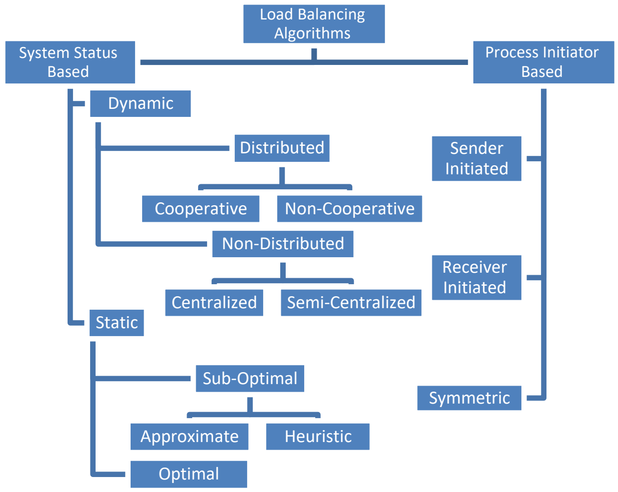

Fig. 3 Taxonomy of load balancing algorithms

4-4- Load balancing in cloud computing

Load balancers are advantageous to cloud environments in which massive workloads overloading a single server is highly likely; hence, many high-level services will be unavailable, and thus, adversely affecting both response time, service reliability, and Service-Level Agreement (SLA)—all of which are critical to business processes.

4-5- Load Balancing in Fog Computing

In fog computing, data sent by IoT devices/sensors is transferred to the fog nodes. Due to the high rate of data generation, some fog nodes get overloaded; hence, a load balancer should be used to offload the tasks to the nodes that are less overloaded [11].

5- Classification of load Balancing Algorithms in Fog and Cloud Computing

Load Balancing (LB) is NP-Pharisees., and finding real solutions for NP-hard algorithms is too costly. Due to the Non-deterministic of this problem, various methods have been used to balance the load among cloud and fog nodes. In this section, as shown in Figure 4, we classify load balancing algorithms into 4 groups: Meta-heuristic algorithms, Heuristic algorithms, Algorithms employing machine learning, and Custom algorithms.

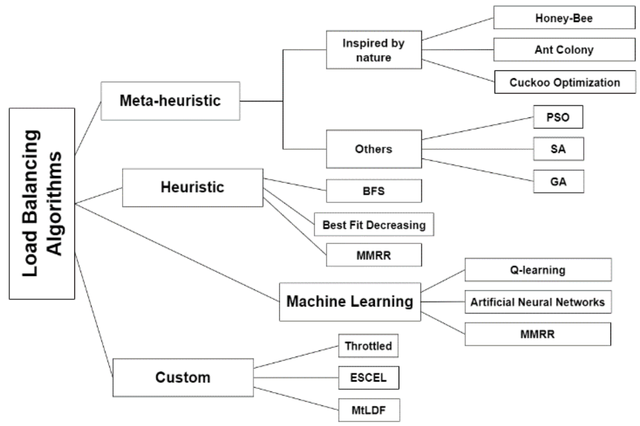

Fig. 4 Classification of load balancing algorithms and some of their examples

5-1- Meta-Heuristic LB Algorithms

In this section, we review several different load balancing strategies. Meta-heuristic methods—taking inspiration from nature or biological behaviors—consider some of the optimization hypotheses meta-heuristic methods, rather than heuristic algorithms, require more time to obtain the final solution. Amongst the meta-heuristic algorithms are the Hill-climbing algorithm, Honey-Bee algorithm, Particle Swarm Optimization (PSO) algorithm, Simulated Annealing (SA) algorithm, Genetic Algorithm (GA), and Ant Colony (ACO) algorithm.

5-2- Classification based on Meta-Heuristic Algorithms in Fog Computing

PSO algorithm, which is a meta-heuristic algorithm, has been used in papers [12], [13], [14]. The authors in [12] applied a new architecture based on SDN networks in the fog/cloud environment called SDCFN to obtain the desired load balance and reduce the distance between intelligent vehicles by virtue of PSO. [13] provides a model based on load balance and energy-aware planning in the fog environment to solve energy consumption issues in smart factories. The PSO algorithm is used to prioritize the workload. [14] presents a fog/cloud-based approach to reduce processing time and response time. It uses PSO-SA algorithms to properly allocate requests (virtual machines) and balance loads between virtual machines. [15] uses four load balancing algorithms, Throttled, PSO, RR, and Active VM Load Balancing (MLB), and four-layered architecture, to manage users' requests for electricity and reduce energy consumption.

[16] proposes a new algorithm for task scheduling with the tasks being modeled with a directed acyclic graph *G(V, E)*, where *V* are the tasks with their respective weights indicating their execution times, and *E* are the prerequisite relations between the tasks, with their respective weights

indicating the communication cost of sending a message between two tasks. The authors use the Genetic algorithm to obtain the initial population as answers to the task scheduling problem, and further, use the PSO algorithm to find the optimal solution. They have used a novel cost function based on these two algorithms to measure task execution time on available resources and their method shortens the length of the critical path and reduces the communication costs among the processors.

The Honey-Bee algorithm is another meta-heuristic algorithm used in papers [17] and [18]. [17] advances a new architecture to balance the traffic load. This load balancing method is decentralized, which helps achieve a load balance between virtual machines in the fog environment. By applying the honey bee algorithm and proposing a new architecture, it allocates resources optimally. [18] have used the Honeybee algorithm to prioritize user requests, minimize energy consumption, and reduce execution time applications.

The Hill-Climbing algorithm, which is in the category of meta-heuristic algorithms, is used in paper [19] to balance the fog computing load by managing the request load from consumers to the appropriate virtual machines. Optimal load balancing is a significant matter in fog computing using tabu search fog computing for load balancing [20]. The paper [21] uses ACO and PSO algorithms to efficaciously distribute load balance among the fog nodes.

5-3- Classification based on Meta-Heuristic Algorithms in Cloud Computing

The Ant Colony algorithm, categorized as a meta-heuristic algorithm, is used in papers [22], [23], [24], [25] to balance the cloud computing load. The paper [22] analyzes the performance of four load balancing algorithms which were inspired by nature, to find data center processing time (DCPT) and total response time (TRT) in the cloud environment. The paper [23] proposed a meta-heuristic approach to the ant colony optimization algorithm. This algorithm solves the task scheduling problem by focusing on minimizing the makespan/computation time only on two objectives. The paper [24] has proposed a modern distributed VM migration strategy named ACO-VMM with high scalability and reliability. Moreover, to find the near-optimal mapping between virtual/physical machines, they propose two approaches inspired by two traversing strategies for ants. The paper [25] advanced a new algorithm based on improved ant colony optimization to ameliorate the process of allocating resources and guarantee the quality of service. The Honey-Bee and Cuckoo Optimization Algorithm (COA) algorithms are the meta-heuristic algorithms used in these papers [26] and [27].

[28] proposes a load balancing algorithm focused on saving energy by mimicking the life of a bird family called cuckoos (COA). Cuckoos raise their young by laying eggs in the roosts of other birds with similar eggs. Using COA, over-utilized hosts are detected, and afterward, some VMs are chosen for migration from these hosts to others. The paper [26] advanced a new efficient load balancing algorithm based on bee colonies, in which the tasks removed from overloaded VMs and under-loaded VMs are regarded as honey bees and food sources, respectively. The effort of this technique is to reduce response time and the number of task migrations.

The paper [27] proposed a novel modified Artificial Bee Colony (ABC) method named Mutation Based ABC (MABC). This algorithm highlights the procedure of detecting under-utilized available servers in the provided data centers. In line with that, the paper [29] introduces the integration of the swarm intelligence algorithm in an artificial bee colony with a heuristic scheduling algorithm named Heuristic task scheduling with Artificial Bee Colony (HABC).

[30] proposes a new version of the meta-heuristic Grey Wolf Optimization (GWO) algorithm, which mimics the hunting behavior of grey wolves, with alpha wolves as leaders, and beta, delta, and omega wolves in the next ranks, forming a hierarchy. The authors model the nodes (VMs) in a cloud infrastructure as preys for a pack of wolves. Using a load threshold and based on estimated loads, they try to find under loaded nodes and recommend them to the server. Their method outperforms PSO, ABC, and GA in terms of makespan, cost, response time, and resource utilization.

The Water Wave Algorithm (WWA) algorithm is another meta-heuristic algorithm used in the paper [31]. The paper aims for resource scheduling in the cloud environment.

The PSO algorithm is another meta-heuristic algorithm used in papers [32] and [33]. The algorithm is based on the heuristic optimization technique and used for analyzing the optimal path of solution space; while putting upload on a specific VM for processing of resources, it moves along all the VM and determines the optimal machine to put the load. The paper [32] introduced a load balancing strategy by using revised PSO task scheduling (LBMPS). The paper [33] proposed a new multi-criteria optimization technique for the weighted task scheduling that is called PSO based αPSO-TBLB (Task Based Load Balancing) load balancing method.

The genetic algorithm, which is a meta-heuristic algorithm, has been used in the paper [34]. The idea behind considering the priority is real-world virtualization. The authors advanced a policy for cloud task scheduling based on the load balancing Enhanced Genetic Algorithm (EGA). This algorithm schedules VMs in a way that load balancing is achieved, and the need for VM migrations is reduced due to its smart way of allocating VMs to physical machines using the fitness function. Table 1 mentions the prominent meta-heuristic solutions to the load balancing problem.

Table 1: Meta-heuristic load balancing algorithms

Author(s)	Objective	Technique	Testbed/Sim.	Target Service
Sefati and Mousavinasab 2022 [30]	Improve makespan utilization and reduce response time	GWO algorithm	Cloudsim	Cloud
Hussein et al. 2020 [21]	Improve response time	ACO and PSO algorithms	MATLAB	Fog
Bukhsh et al. 2018 [14]	Improve response, processing, and execution time	PSO and Simulated Annealing algorithms	Cloud Analyst	Fog
Zahid et al. 2018 [19]	Improve response time, processing time, and delay	Hill-Climbing algorithm	Cloud Analyst & Java	Fog
Abbasi et al. 2018 [15]	Improve response time and delay	Round Robin, PSO, Throttled, and Active VM load balancing algorithms	Cloud Analyst	Fog
Arulkumar and N. Bhalaji 2020 [22]	Improve TRT and DCPT	ACO, PSO, GA, and WWA algorithms	Cloud Analyst	Cloud
Kruekaew and Kimpan 2020 [29]	Maximize productivity and minimize total makespan	ABC and heuristic scheduling algorithms	Cloudsim	Cloud
Sharma and Saini 2019 [18]	Minimize energy consumption and execution time	Honey-bee algorithm	MATLAB	Fog
Alguliyev et al. 2019 [33]	Minimize the task execution and transfer time	PSO algorithm	Cloudsim & JSwarm	Cloud
Tellez et al. 2018 [20]	Minimize memory consumption and computational costs	Tabu-search algorithm	Cloudsim	Fog

5-4- Classification based on Heuristic Algorithms

Heuristic methods are a collection of constraints aimed at finding a suitable solution to a specific problem. Heuristic algorithms offer an approximate solution to the best solution.

5-5- Classification based on Heuristic Algorithms in Fog Computing

The Breadth-First Search (BFS) and Best Fit Decreasing (BFD), which are heuristic algorithms, are used in the papers [35] and [36], respectively. [35] proposed a secure method for load balancing and assigning tasks in edge data centers (EDC). Edge data centers are placed midst the cloud data centers and reduce network congestion, and delay by processing user requests and data in a near real-time—breadth-first search algorithm deployed to balance the workload. In this paper, the major objective is to load balancing between different types of computational nodes. First, [36] proposed a model for load balance in the fog/cloud setting. They considered a heuristic method for proper planning and location of virtual machines with virtual machine migration.

The Min-Min and the Max-Min algorithms are static load balancing algorithms classified as Heuristic algorithms. The paper [37] uses this algorithm. The authors proposed a central load balancing policy in the fog computing setting. In this paper, a Min-Min algorithm, a simple and easy algorithm, is used to balance the load of requests. Resources are classified as reliable and unreliable in the fog layer. The paper [38] proposes three heuristic algorithms that carry out load balancing among Micro Data Centers (MDCs): minimum load, minimum distance, and Minimum Hop Distance and Load (MHDL).

5-6- Classification based on Heuristic Algorithms in Cloud computing

As mentioned, Min-Min and Max-Min are static load balancing algorithms which belong to the class of heuristic algorithms. The papers [39], [40], [41] use these algorithms.

In the article [39], Min-Min and Max-Min load balancing algorithms were analyzed. The Min-Min algorithm prioritizes tasks with smaller resource demands and minimum completion times when allocating the resources.

The paper [40] proposed a new load balancing algorithm, which combines Max-Min and Round-Robin algorithm (MMRR) to assign virtual machines to different userbase requests.

The authors in [42] propose a solution to load balancing in big data applications performed on clouds. They provide two mathematical optimization models, one to find a host machine with the maximum number of available resources, and another, for task scheduling. With the aim of reducing execution response time, their load balancer, based on the Hill-climbing algorithm, carries out resource allocation and task scheduling. The key point of their solution is considering a deadline in model optimization for task scheduling and execution that distinguishes the proposed algorithm from existing ones. Their solution transcends FIFO, Round-Robin, MET, Min-Min, Max-Min, Genetic, ESCE, and Throttled algorithms in response time and turnaround time.

The paper [41] proposed a Max-Min scheduling algorithm. The proposed MMSIA algorithm uses the "learned learning" machine learning to improve requests' completion time by clustering requests' sizes and the utilization percent of VMs. Table 2 mentions important heuristic load balancing approaches.

Table 2 : Heuristic load balancing algorithms

Author(s)	Objective	Technique	Testbed/Sim.	Target Service
Aghdashi and Mirtaheri 2021 [42]	Reduce execution response time	Hill-climbing algorithm	Cloudsim	Cloud
Moses et al. 2020 [40]	Improve response time and cost-effectiveness	Max-Min algorithm	Cloudsim	Cloud
Singh and Auluck 2019 [38]	Improving response time	MHDL algorithm	iFogSim	Fog
Hung et al. 2019 [41]	Improve completion time	Max-Min algorithm	Cloudsim	Cloud
Xu et al. 2018 [36]	Improve load balance among computational nodes	BFD algorithm	Cloudsim	Fog
Manju and Sumathi 2018 [37]	Improve response time	Min-Min algorithm	Cloud Analyst	Fog
Puthal et al. 2018 [35]	Improve delay and response time	BFS algorithm	MATLAB	Fog
Gopinath and Vasudevan 2015 [39]	Improve makespan	Min-Min & Max-Min algorithms	Cloudsim	Cloud

5-7- Classification based on Machine Learning Algorithms

Using machine/deep learning (neural networks) techniques, we can obtain accurate predictions with data trained in different situations and virtual machines in cloud and fog environments. It is also possible to host a virtual machine in a much shorter time. Among the methods used in this type of technique, we can mention KNN, Q-learning, ANN, and so on.

5-8- Classification based on Machine Learning Algorithms in Fog Computing

Q-learning algorithm, which is one of the machine learning techniques, is used in the paper [43] to improve response time, delay, and energy consumption. An algorithm is needed to balance the load due to the uncertainty related to user requests and different computing capacities—I have used an algorithm based on reinforcement learning. As mentioned, the technique of artificial neural networks, which is one of the methods based on machine learning, has been used in [44]. The authors in have used a four-layer architecture to minimize delays and energy consumption, load balance, and optimally assign and schedule the task in the fog environment.

5-9- Classification based on Machine Learning Algorithms in Cloud Computing

Clustering or cluster analysis, which is one of the machine learning techniques, is used in the papers [45], [46], [47], [48], [49]. In the paper [45], a new heuristic method named LB-BC (Load Balancing based on Bayes and Clustering) is proposed. The LB-BC method uses the Bayes theorem to acquire the posterior probabilities of every candidate physical host.

The article [46] advanced an algorithm for cluster-based load balancing that performs adequately in heterogeneous node environments. This algorithm takes into consideration the tasks' resource-specific requirements and reduces the overhead cost of scanning by dividing the machines into clusters.

The article [47] introduces an algorithm able to provide more fine-tuned analytical data using machine learning methods, which can form the load scheduling mechanism. The algorithm is based on dynamic load balancing.

The paper [48] presents a method for accelerating the training of a distributed machine learning model based on a cloud service. The authors proposed a load balancing method called fast adaptive reassignment (AdaptFR). The paper [49] proposed a strategy based on a machine-learning algorithm for intelligent VM scheduling that tries to attain load balancing of the cloud data center. Table 3 shows the load balancing approaches which employ machine learning.

Table 3 : Machine learning load balancing algorithms

Author(s)	Objective	Technique	Testbed/Sim.	Target Service
Baek et al. 2019 [43]	Minimize latency, response time, and overload (extra cost)	Q-learning algorithm, Using three load transfer models for testing	-	Fog
Sharma and Saini 2019 [44]	Improve response time, latency, energy consumption, load balance rate	Artificial Neural Networks (ANN)	iFogSim	Fog
Parida and Panchal 2018 [47]	More efficient load balancing	Dynamic load balancing	AWS	Cloud
Zhao et al. 2016 [45]	Reduce failed number of task deployment events, improve throughput and performance of external cloud services	Naïve Bayes classification and Clustering	Cloudsim	Cloud
Kapoor and Debas 2015 [46]	Improve waiting time, execution time, turnaround time, and throughput	k-Means clustering algorithm	Java	Cloud

5-10- Classification based on Custom Algorithms

Custom algorithms are the proposed algorithms by authors based on innovative models. By studying the load balancing algorithms in fog and cloud, we have faced proposed algorithms that are not based on the known models, and the model is innovated. To continue, we will mention these researches in fog and cloud computing.

5-11- Classification based on Custom Algorithms in Fog Computing

Categories in fog computing are based on customized algorithms, algorithms, or strategies written by the paper authors to improve standards such as improved latency, response time, power consumption, energy consumption, and the like. This category includes techniques such as First-In-First-Out (FIFO), Throttled, Equally Spread Current Execution Load (ESCEL), Min-Min, and Max-Min.

The authors in [50] advance the MOABCQ method, which is a multi-objective task scheduling approach using hybrid artificial bee colony algorithm along with Q-learning. Their method calculates the fitness of the VMs, based on which, considers the selection of them. The MOABCQ method improves throughput, cost reduction, makespan reduction, and resource utilization.

[51] aims to process and prioritize input requests using the queue model under the SLA law. To establish a strategy for allocating resources, [36] introduces a dynamic resource allocation method named DRAM in the fog network. The introduced technique consists of four main parts for load balancing among nodes in the cloud and fog platforms. DRAM's implementation is such that it allocates the resources statically and schedules them in a dynamic manner in fog services through identifying the spare spaces, global resource allocation based on load balance, partitioning the fog service, and static resource allocation for the subsets of the fog service.

The authors in [52] aim to reduce energy consumption, cost, and time by making appropriate decisions and scheduling load transfer among fog nodes. To optimize and distribute the load in fog settings by taking into account specific multi-tenancy demands (priority and delay), the authors in [53] proposed the Multi-tenant Load Distribution algorithm for Fog Environments (MtLDF).

The authors in [54] proposed an algorithm for load balancing in fog computing focused on graph partitioning. In their paper, the physical node graph model is viewed as a VM graph model. Afterward, depending on the resource distance and task load balancing, using a graph partition and clustering, the VM node provides services to the user.

Fog-Based Radio Access Networks (F-RAN) have an important role in future 5th generation (5G) cellular networks. [55] introduced the concept of virtual FAPs (v-FAPs), set up by several local IoT devices under the control of the FAPs. In this paper, the first authors formulated an optimization problem for optimal task assignment to reduce the maximum resource costs. Then they present a service load balancing algorithm for the v-FAPs to assign appropriate tasks.

Increasing the traffic load in healthcare systems causes all the requests to be sent to the main server to be delayed.

Delays are intolerable in healthcare scenarios. To alleviate this issue, the authors of [56] aim to provide efficient resource utilization by conjugating fog computing support so that the requests are dealt with by foglets, and only crucial requests are sent to the cloud to be processed.
The authors in [57] proposed an adaptive load balancing algorithm called LBA-le (Load Balancing Algorithm for IoT communications within e-health environment). The proposed load-balancing algorithm is based on integrating IoT communication parameters in the flow control process supported by the TCP protocol to consider the network fluctuations and apply them in the e-health domain as well.
As the demand for numerous IoT applications increases, fog nodes tend to overload, even close to the sensors; hence the response time of IoT applications and latency increases. As a result, [58] proposed an algorithmic model that takes into consideration the dynamics and heterogeneity of computational nodes in fog computing. These models utilize the predefined policies by the network administrator to assign tasks to the most fitting nodes.
The authors in [59] used comprehensive dynamic resource allocation for load balancing. The method used in this paper includes the following four phases:
1. Service partitioning. 2. Gathering spare space details. 3. Primary static resource allocation. 4. Dynamic resource allocation that secures global load-balancing in the fog environment.
In cloud registering, Load-adjusting is one of the testing undertakings. Various load balancing strategies are proposed for load adjusting. The authors in [60] proposed a heap adjusting calculation. Load adjusting—a dynamic strategy—is the system to adjust the heap to the cloud hubs so that Computing Communication and Signal processing in every hub viably uses the assets and limits the reaction time.
The authors in [61] introduced two new load sharing mechanisms, such as adaptive forwarding, and sequential forwarding, to offload tasks towards the neighboring nodes.
The authors in [62] proposed a load balancing technique for IoT-Gateways and network links through the use of Software-Defined Networks (SDN). The main goal of this method is performance improvement in IoT scenarios based on fog computing. The authors in [63] proposed LL(F, T) power-of-random choices based on distributed peer-to-peer load balancing algorithm.

5-12- Classification based on Custom Algorithms in Cloud Computing

The paper [64] proposed algorithm-based Self-learning and Adaptive Load balancer (SSAL). The algorithm focuses on data centers' overall throughput optimization in unstable environments. In order to estimate the recent capabilities of the servers and assign workloads commensurate to the current relative potential of the servers, SSAL logically splits the time into fixed-length feedback intervals.
The paper [65] introduces a hybrid strategy for load balancing and task scheduling called Dems. The strategy embodies three main algorithms: Querying and Migrating tasks (QMT), On-Demand scheduling, and Staged Task Migration (STM).
The paper [66] aims to enhance the performance of the computing clusters by advancing a combination of centralized and decentralized load balancing. In the proposed load balancing algorithm, computing nodes notify other neighboring nodes of their load and resource usage details to determine their relative state. The resource availability information and load status of all the nodes in each cluster, based on which workload distribution and migration come about, are stored in the main node of the cluster.
The paper [67] aims to optimize the load and schedule resources for each cloud user request with the efficient transformation of the data center by proposing the Fuzzy-based Multidimensional Resource Scheduling and Queuing Network (F-MRSQN) method. The method's major intent is to effectively put integrated scheduling and load balancing algorithms into use, depending on minimum processing time and maximum resource utilization in the cloud environment. This method's main objective is to effectively utilize combined scheduling and load balancing algorithms based on maximum resource usage and minimum processing time in the cloud environment.
The paper [68] presented a new hybrid load balancing algorithm, an amalgamation of randomizing and greedy load balancing algorithms. The main goal is to improve the response time for the user (UserBase) and the processing time of the data center.
The paper [69] advanced a novel mechanism for load balancing. This method is used for calculating server processing power. It is also able to load and obtain PS values, thus reducing the chance of a server being incapable of handling excessive computational requirements.
The authors in [70] advance a model in which, without a central node to manage the system load, each individual node is responsible to estimate its status based on its computing power and the intended volume of load, which will classify themselves into nodes with positive load, the nodes with less computing power relative to their considered load, and nodes with negative load, which will undertake extra portions of the positive nodes' load, leading to load balance. They also define a parameter entitled compensating factor, to address communication delay between nodes, which is calculated from each node's perspective, and to compensate the effect of external load by using information from neighbor nodes status. Their simulation results illustrates significant improvement in comparison with common distributed load balancing approaches in managing dynamic requests.

In the paper [71], a load balancing algorithm based on VM availability is proposed. In particular, the Availability Index (AI) is assessed for all VMs over a specific period, based on which the jobs are allotted to the machines. Table 4 mentions some of recent custom load balancing approaches.
[72] presents a new mechanism for flow scheduling in cloud data centers. This method makes decisions based on flows' sizes. Small flows (mice) are sent via the EMCP algorithm and big flows (elephant) are scheduled using bidirectional search. Their approach can balance the network more efficiently than traditional Static, ECMP, and DiFS mechanisms.

5-13- Load Balancing Algorithms in Converged Fog and Cloud Computing

The article [36] advanced a Virtual Machine scheduling approach for load balancing in fog/cloud computing. By exploiting the VM live migration method, the authors design a VM scheduling mechanism through dynamic VM scheduling and VM placement.
The paper [74] introduces a new mechanism for scheduling IoT requests using a made-to-order implementation of the genetic algorithm (GA) as a heuristic procedure that mainly aims to improve the overall latency. The authors study the GA and evaluate it on different problems with various sizes to estimate the effects of the model with different parameters, namely the maximum number of iterations or the population size.
The paper [73] advanced an energy-efficient load balancing mechanism for scientific workflows in the fog/cloud computing environment along with a load balancing algorithm for the fog environment. Load balancing at the fog layer facilitates latency reduction, improving the quality of service, and using the resources properly. The mechanism aims to utilize the resources at the fog layer by minimizing the energy consumed by fog resources.
The authors of [75], proposed different algorithms for load balancing, task scheduling, and resource provisioning, and they recognized some of their drawbacks for further development. They surveyed the fog-integrated cloud environment and its 3-layered architecture in their paper.
The paper [76] advanced a Fault-Tolerant Scheduling Method (FTSM) for distributing service requests to ample devices in IoT-based fog/cloud environments. The method mainly aims to increase the capacity along with reliability and reduce the overhead costs and latency of cloud services.
The paper [77] has proposed a reliable scheduling approach, named the Load Balanced Service Scheduling Approach (LBSSA), for allocating users' requests to the resources of cloud-fog environments. LBSSA mainly aims to achieve proper system utilization, high load balancing, and reliable service for requests within the necessary limits of response times.

6- Analysis of Research

In this article, different load balancing algorithms in the cloud, fog computing, and convergence environment are surveyed and compared. According to the review of selected papers, we give different analyses based on different categories, algorithm types, the objectives of load balancing algorithms, task specifications, selecting a suitable location for task execution, and simulation tools.
The first categorization of the research on the load balancing algorithms is based on their type and approach. We categorize the proposed algorithms into metaheuristic, heuristic, machine learning, and customized algorithms. The results show that custom algorithms are proposed by most of the authors for fog computing load balancing algorithms. In contrast, most authors use meta-heuristic algorithms to balance the load in the cloud environment. The number of researches on proposing load balancing algorithms for converged fog and cloud computing environments is not too many yet, and it's a new scope of research for researchers in the load-balancing field.
In the converged fog and cloud computing, we face two different architectures, distributed nature of the fog computing management model and the centralized nature of the cloud computing management model. Since the load balancing algorithm is contingent upon the architecture of the system, it's a very interesting issue that we can propose a load balancing algorithm in the converged system to balance the load in all cloud and fog nodes. Based on the architectures, there are new issues such as designing distributed load balancing algorithms or centralized or the hybrid model.
By categorizing the research based on the type of algorithms, in the second level in fog and cloud, respectively, we can mention meta-heuristic algorithms in fog computing and custom algorithms in cloud computing, and in the third level, heuristic algorithms in fog and machine learning in cloud environment have been more popular. Finally, in the fourth level, the machine learning algorithms in fog and heuristic algorithms in cloud computing have been used. This issue is depicted using Figure 4 and Figure 5.

Table 4 : Custom load balancing algorithms

Author(s)	Objective	Technique	Testbed/Sim.	Target Service
Kruekaew and Kimpan 2022 [50]	Optimize task scheduling, maximize VM throughput, create load balance based on makespan	ABC and Q-learning algorithms	Cloudsim	Cloud
Kaur and Aron 2021 [73]	Reduce energy consumption, optimize resource utilization, improve QoS	Energy-aware load balancing framework	iFogSim	Cloud & Fog
Beraldi et al. 2020 [61]	Improve response time	Sequential forwarding and adaptive forwarding algorithms	MATLAB & Omnet++	Fog
Khattak et al. 2019 [56]	Improve latency and reduce traffic overhead by improving QoS	An algorithm to measure patients' heart condition, along with an algorithm for using fog servers	iFogSim & C	Fog
Mirtaheri and Grandinetti 2016 [70]	Reducing the search space in the load balancing problem, via a novel decentralized approach	A new decentralized model for estimating each node's status and load assignment accordingly	MATLAB	Cloud

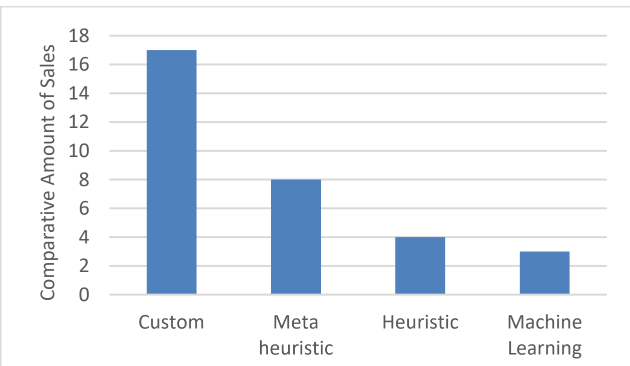

Fig. 5 Load balancing algorithms in fog computing

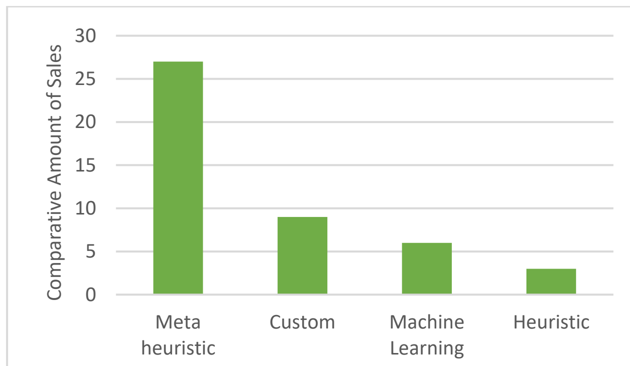

Fig. 6 Load balancing algorithms in cloud computing

In accordance with this categorization, it can be concluded that the famous algorithms have not succeeded yet in solving the load balancing issue in fog computing environments. But on the other hand, metaheuristic algorithms are the most widely used in providing load balancing algorithms in cloud computing environments. For proposing a suitable algorithm to work efficiently in converged fog and cloud computing environments, this categorization can help.

The second categorization of research is done based on the objective of the proposed load balancing algorithms in these researches. In fog computing, response time, execution time, processing time, delay and latency, throughput, computational cost, energy consummation, overload, resource utilization, power consummation, and failure rate are the most popular objectives in balancing the load of the system. Through these objectives, response time is in the first rank, and minimizing the delay and latency in the second rank is the most popular research objective. The statistic chart is shown in Figure 6.

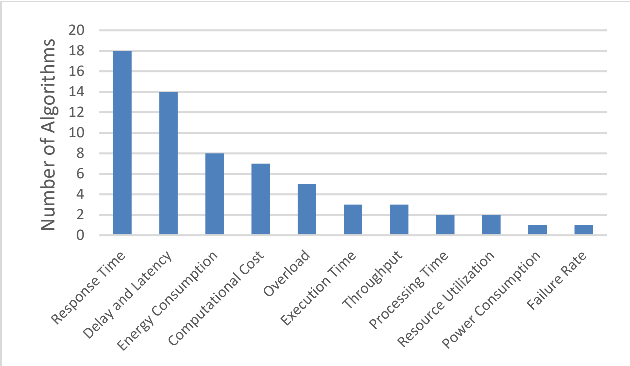

Fig. 7 Objective of load balancing algorithms in fog computing

In cloud computing, Response Time, Execution Time, Processing Time, Waiting Time, Throughput, Computational Cost, Energy Consumption, Completion Time, Makespan, Migration, Accuracy, Resource Utilization, Power Consummation, and Failure rate are the most popular objectives in designing load balancing platform. As mentioned in Figure 7, resource utilization is in the first rank in objectives of research and makespan state in the second rank. It should be noted that the proposed approaches to bring solutions to the load balancing problem commonly use random, Google Cloud Jobs (GoCJ), and synthetic workloads as their datasets for evaluating the above criteria.

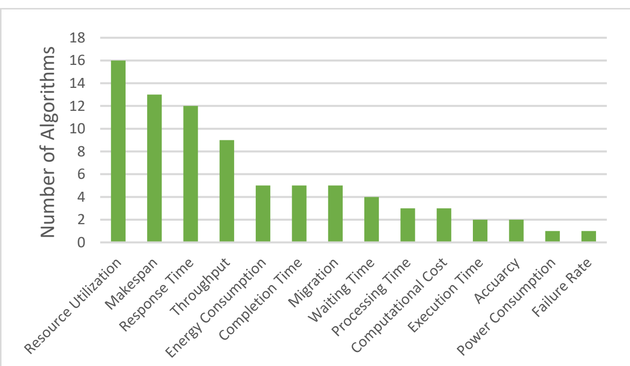

Fig. 8 Objective of load balancing algorithms in cloud computing

Based on these statistics, the objective of balancing the load in the converged fog and cloud computing settings will be a critical challenge. It's possible to select a single objective for the whole in the system or have a separate objective for the fog and cloud environment. For example, the load balancing strategy in distributing the fog-based task can be based on minimizing the response time, and the strategy of load balancing for the cloud-based tasks can improve resource utilization.

The third analysis is based on the task specification. The fog-based tasks have special specifications, and also, the cloud-based tasks have their specifications. The load balancer should prioritize the tasks based on their specifications to reach the optimum output. Real-time nature of tasks, the priority of the tasks regarding the user's request, size of tasks, runtime duration of the task, computing-intensive tasks, data-intensive tasks, the location of requested data by tasks, the I/O requested tasks, etc. These are the challenges that the load balancer should be able to consider to manage the tasks. Therefore, if we have an environment with fog and cloud services, we should be able to consider the specification of tasks in deciding to run the tasks.

The fourth categorization of issues is about selecting a suitable location for task execution. The load balancer should be informed about the nodes' statuses in the system to find the location. There are different strategies for obtaining this information. However, we can categorize them into two strategies, pooling and interrupting. Pooling means that the load balancer in different periods asks the nodes to send the status of CPU and memory or any other needed information from the node to estimate the load of the system and madding decisions about migrating the tasks to other nodes or not. Interrupting strategy means that the nodes send the overloading alarm to the load balancer when the resource utilization rate reaches the specified and predefined rate. By receiving this alarm, the load balancer makes a new decision to migrate the tasks to another suitable node to run.

Statistically, Figure 8 and Figure 9 show the percentage of evaluation tools used to review the literature here in this paper. The CloudAnalyst, iFogSim, and Cloudsim have 4.11% each, MATLAB has 7.20% of usage, C#, Java platform, JMeter, Mininet, Python 3.7, Simply package come next evaluation tools for these literature reviews in fog computing. But, Cloudsim has 67% of usage in cloud computing. Then CloudAnalyst has 14% usage, Java platform, AWS, MATLAB, and Rock Cluster come next evaluation tools for these literature reviews in cloud computing. Some aspects to consider while choosing these tools include the ability to create quantities with different random distributions, providing reports on the infrastructure's performance in the form of figures and curves, and user support, i.e., timely updates, and informative documentation [78].

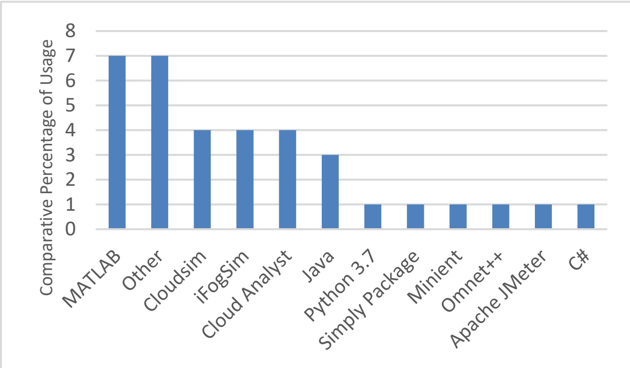

Fig. 9 Evaluation tools for load balancing mechanisms in fog computing

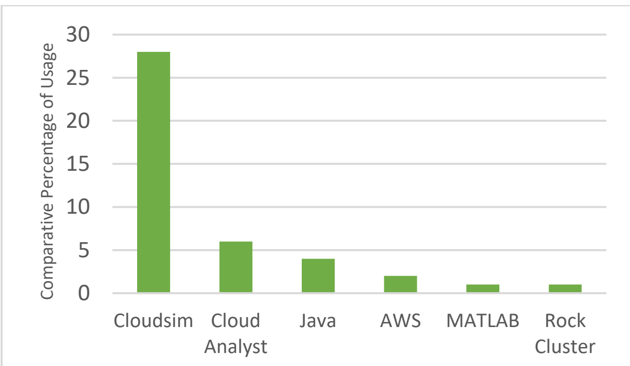

Fig. 10 Evaluation tools for load balancing mechanisms in cloud computing

7- Conclusion and Future Work

Cloud is an enormous system with various aspects, which involves cloud service providers, myriad end-users, service brokers, physical hardware machines, storage capabilities, bandwidth, internet latency, storage capabilities, scheduling algorithms, etc. Fog-based processing is a processing pattern as a result of the rapid advancement of the Internet of Things (IoT), central processing systems, and mobile internet. This processing model responds efficiently to the needs of real-time and delay-sensitive applications. The load balancer is one of the most important issues in the fog and cloud calculation model because overloading the system will reduce efficiency. Therefore, efficient algorithms are needed to load balance, optimal resource allocation, reduce response time, and increase system efficiency.

This review paper discusses new load-balancing algorithms in the fog and cloud and their converged environment. According to the classification, the most common categories of papers written in load balancing in fog and cloud computing, respectively, are custom and meta-heuristic algorithms. For designing suitable computing algorithms in a converged fog and cloud environment, the presented categories will be useful and give a better understanding of the solutions ahead.

Further research may include considering more aspects of QoS such as security, delay for different routing policies, fault tolerance and etc. With the ongoing advancements in the discipline of artificial intelligence, utilizing optimization techniques along with other machine learning algorithms should also be considered. Taking inspiration from nature has been in the spotlight to bring solutions to the problem of load balance, nevertheless, more nature-inspired algorithms can be developed. Moreover, taking into account green computing, where its usage saves energy and reduces the trade-off between SLA requirements and the energy consumed, is suggested. Furthermore, with 5G networks becoming more and more prevalent, considering their capabilities and flaws can significantly affect future research conducted to solve this problem. Finally, as the progress of research in this area suggest, in the future, the load balancing will be carried out dynamically, and with special focus on the dependent tasks.

References

[1] P. Hu, S. Dhelim, H. Ning, and T. Qiu, “Survey on fog computing: architecture, key technologies, applications and open issues,” Journal of Network and Computer Applications, vol. 98, pp. 27–42, Nov. 2017, doi: 10.1016/j.jnca.2017.09.002.

[2] M. Verma, N. Bhardwaj, and A. K. Yadav, “Real Time Efficient Scheduling Algorithm for Load Balancing in Fog Computing Environment,” International Journal of Information Technology and Computer Science, vol. 8, no. 4, pp. 1–10, Apr. 2016, doi: 10.5815/ijitcs.2016.04.01.

[3] S. Verma, A. K. Yadav, D. Motwani, R. S. Raw and H. K. Singh, "An efficient data replication and load balancing technique for fog computing environment," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 2888-2895.

[4] A. Alharthi, F. Yahya, R. J. Walters, and G. B. Wills, “An Overview of Cloud Services Adoption Challenges in Higher Education Institutions,” Proceedings of the 2nd International Workshop on Emerging Software as a Service and Analytics, 2015, doi: 10.5220/0005529701020109.

[5] S. Mathew, “Implementation of Cloud Computing in Education - A Revolution,” International Journal of Computer Theory and Engineering, pp. 473–475, 2012, doi: 10.7763/ijcte.2012.v4.511.

[6] H. Atlam, R. Walters, and G. Wills, “Fog Computing and the Internet of Things: A Review,” Big Data and Cognitive Computing, vol. 2, no. 2, p. 10, Apr. 2018, doi: 10.3390/bdcc2020010.

[7] A. Ahmed, et al. "Fog Computing Applications: Taxonomy and Requirements," arXiv, 26 July 2019. doi: 10.48550/arXiv.1907.11621.

[8] E. Jafarnejad Ghomi, A. Masoud Rahmani, and N. Nasih Qader, “Load-balancing algorithms in cloud computing: A

survey," Journal of Network and Computer Applications, vol. 88, pp. 50–71, Jun. 2017, doi: 10.1016/j.jnca.2017.04.007.

[9] Z. M. Elngomi and K. Khanfar "A Comparative Study of Load Balancing Algorithms: A Review Paper," International Journal of Computer Science and Mobile Computing, vol. 5, no. 6, pp. 448-458, Jun. 2016.

[10] M. N. Arab, S. L. Mirtaheri, E. M. Khaneghah, M. Sharifi, and M. Mohammadkhani, "Improving Learning-Based Request Forwarding in Resource Discovery through Load-Awareness," Lecture Notes in Computer Science, pp. 73–82, 2011, doi: 10.1007/978-3-642-22947-3_7.

[11] A. Chandak and N. K. Ray, "A Review of Load Balancing in Fog Computing," 2019 International Conference on Information Technology (ICIT), Dec. 2019, doi: 10.1109/icit48102.2019.00087.

[12] X. He, Z. Ren, C. Shi, and J. Fang, "A novel load balancing strategy of software-defined cloud/fog networking in the Internet of Vehicles," China Communications, vol. 13, no. 2, pp. 140–149, 2016, doi: 10.1109/cc.2016.7405730.

[13] J. Wan, B. Chen, S. Wang, M. Xia, D. Li, and C. Liu, "Fog Computing for Energy-Aware Load Balancing and Scheduling in Smart Factory," IEEE Transactions on Industrial Informatics, vol. 14, no. 10, pp. 4548–4556, Oct. 2018, doi: 10.1109/TII.2018.2818932.

[14] R. Bukhsh, N. Javaid, Z. Ali Khan, F. Ishmanov, M. Afzal, and Z. Wadud, "Towards Fast Response, Reduced Processing and Balanced Load in Fog-Based Data-Driven Smart Grid," Energies, vol. 11, no. 12, p. 3345, Nov. 2018, doi: 10.3390/en11123345.

[15] 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," Lecture Notes on Data Engineering and Communications Technologies, pp. 737–750, Oct. 2018, doi: 10.1007/978-3-030-02613-4_66.

[16] A. Kamalinia and A. Ghaffari, "Hybrid Task Scheduling Method for Cloud Computing by Genetic and PSO Algorithms," Journal of Information Systems and Telecommunication (JIST), vol 16, no. 4, pp. 1-10, Sep. 2016, doi: 10.7508/jist.2016.04.008.

[17] M. Verma and N. B. A. K. Yadav, "An architecture for load balancing techniques for fog computing environment," International Journal of Computer Science and Communication, vol. 8, no. 2, pp. 43–49, 2015.

[18] S. Sharma and H. Saini, "Efficient Solution for Load Balancing in Fog Computing Utilizing Artificial Bee Colony," International Journal of Ambient Computing and Intelligence, vol. 10, no. 4, pp. 60–77, Oct. 2019, doi: 10.4018/ijaci.2019100104.

[19] M. Zahid, N. Javaid, K. Ansar, K. Hassan, M. KaleemUllah Khan, and M. Waqas, "Hill Climbing Load Balancing Algorithm on Fog Computing," Advances on P2P, Parallel, Grid, Cloud and Internet Computing, pp. 238–251, Oct. 2018, doi: 10.1007/978-3-030-02607-3_22.

[20] N. Tellez, M. Jimeno, A. Salazar, and E. D. Nino-Ruiz, "A Tabu Search Method for Load Balancing in Fog Computing," International journal of artificial intelligence, vol. 16, pp. 106–135, 2018.

[21] M. K. Hussein and M. H. Mousa, "Efficient Task Offloading for IoT-Based Applications in Fog Computing Using Ant Colony Optimization," IEEE Access, vol. 8, pp. 37191–37201, 2020, doi: 10.1109/access.2020.2975741.

[22] V. Arulkumar and N. Bhalaji, "Performance analysis of nature inspired load balancing algorithm in cloud environment," Journal of Ambient Intelligence and Humanized Computing, Jan. 2020, doi: 10.1007/s12652-019-01655-x.

[23] A. Gupta and R. Garg, "Load Balancing Based Task Scheduling with ACO in Cloud Computing," 2017 International Conference on Computer and Applications (ICCA), Sep. 2017, doi: 10.1109/comapp.2017.8079781.

[24] W. T. Wen, C. D. Wang, D. S. Wu, and Y. Y. Xie, "An ACO-based Scheduling Strategy on Load Balancing in Cloud Computing Environment," 2015 Ninth International Conference on Frontier of Computer Science and Technology, Aug. 2015, doi: 10.1109/fcst.2015.41.

[25] P. Verma, S. Shrivastava, and R. K. Pateriya, "Enhancing load balancing in cloud computing by ant colony optimization method," International Journal of Computer Engineering in Research Trends, vol. 4, no. 6, pp. 277–284, 2017.

[26] K. R. Remesh Babu and P. Samuel, "Enhanced Bee Colony Algorithm for Efficient Load Balancing and Scheduling in Cloud," Advances in Intelligent Systems and Computing, pp. 67–78, Dec. 2015, doi: 10.1007/978-3-319-28031-8_6.

[27] S. Singhal, "Load Balancing in Cloud Computing using Mutative Bacterial Foraging Optimization," Journal of Xidian University, vol. 14, no. 6, Jun. 2020, doi: 10.37896/jxu14.6/258.

[28] M. Yakhchi, S. M. Ghafari, S. Yakhchi, M. Fazeli, and A. Patooghi, "Proposing a load balancing method based on Cuckoo Optimization Algorithm for energy management in cloud computing infrastructures," 2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO), May 2015, doi: 10.1109/icmsao.2015.7152209..

[29] B. Kruekaew and W. Kimpan, "Enhancing of Artificial Bee Colony Algorithm for Virtual Machine Scheduling and Load Balancing Problem in Cloud Computing," International Journal of Computational Intelligence Systems, vol. 13, no. 1, p. 496, 2020, doi: 10.2991/ijcis.d.200410.002.

[30] S. Sefati, M. Mousavinasab, and R. Zareh Farkhady, "Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: performance evaluation," The Journal of Supercomputing, May 2021, doi: 10.1007/s11227-021-03810-8.

[31] A. V and N. Bhalaji, "Load balancing in cloud computing using water wave algorithm," Concurrency and Computation: Practice and Experience, vol. 34, no. 8, Sep. 2019, doi: 10.1002/cpe.5492.

[32] A. Pradhan and S. K. Bisoy, "A novel load balancing technique for cloud computing platform based on PSO," Journal of King Saud University - Computer and

Information Sciences, Oct. 2020, doi: 10.1016/j.jksuci.2020.10.016.

[33] R. M. Alguliyev, Y. N. Imamverdiyev, and F. J. Abdullayeva, "PSO-based Load Balancing Method in Cloud Computing," Automatic Control and Computer Sciences, vol. 53, no. 1, pp. 45–55, Jan. 2019, doi: 10.3103/s0146411619010024.

[34] H. Sharma and G. Sekhon, "Load Balancing in Cloud Using Enhanced Genetic Algorithm," International Journal of Innovations and & Advancement in Computer Science, vol. 6, no. 1, pp. 100–107, 2017.

[35] D. Puthal, M. S. Obaidat, P. Nanda, M. Prasad, S. P. Mohanty, and A. Y. Zomaya, "Secure and Sustainable Load Balancing of Edge Data Centers in Fog Computing," IEEE Communications Magazine, vol. 56, no. 5, pp. 60–65, May 2018, doi: 10.1109/mcom.2018.1700795.

[36] X. Xu, Q. Liu, L. Qi, Y. Yuan, W. Dou, and A. X. Liu, "A Heuristic Virtual Machine Scheduling Method for Load Balancing in Fog-Cloud Computing," 2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS), May 2018, doi: 10.1109/bds/hpsc/ids18.2018.00030.

[37] A. B. Manju and S. Sumathy, "Efficient Load Balancing Algorithm for Task Preprocessing in Fog Computing Environment," Smart Intelligent Computing and Applications, pp. 291–298, Nov. 2018, doi: 10.1007/978-981-13-1927-3_31.

[38] A. Singh and N. Auluck, "Load balancing aware scheduling algorithms for fog networks," Software: Practice and Experience, Jun. 2019, doi: 10.1002/spe.2722.

[39] P. P. G. Gopinath and S. K. Vasudevan, "An In-depth Analysis and Study of Load Balancing Techniques in the Cloud Computing Environment," Procedia Computer Science, vol. 50, pp. 427–432, 2015, doi: 10.1016/j.procs.2015.04.009.

[40] A. Kazeem Moses, A. Joseph Bamidele, O. Roseline Oluwaseun, S. Misra, and A. Abidemi Emmanuel, "Applicability of MMRR load balancing algorithm in cloud computing," International Journal of Computer Mathematics: Computer Systems Theory, vol. 6, no. 1, pp. 7–20, Dec. 2020, doi: 10.1080/23799927.2020.1854864.

[41] T. C. Hung, L. N. Hieu, P. T. Hy, and N. X. Phi, "MMSIA," Proceedings of the 3rd International Conference on Machine Learning and Soft Computing - ICMLSC 2019, 2019, doi: 10.1145/3310986.3311017.

[42] A. Aghdashi and S. L. Mirtaheri, "Novel dynamic load balancing algorithm for cloud-based big data analytics," The Journal of Supercomputing, Aug. 2021, doi: 10.1007/s11227-021-04024-8.

[43] J. Baek, G. Kaddoum, S. Garg, K. Kaur, and V. Gravel, "Managing Fog Networks using Reinforcement Learning Based Load Balancing Algorithm," 2019 IEEE Wireless Communications and Networking Conference (WCNC), Apr. 2019, doi: 10.1109/wcnc.2019.8885745.

[44] S. Sharma and H. Saini, "A novel four-tier architecture for delay aware scheduling and load balancing in fog environment," Sustainable Computing: Informatics and Systems, vol. 24, p. 100355, Dec. 2019, doi: 10.1016/j.suscom.2019.100355.

[45] J. Zhao, K. Yang, X. Wei, Y. Ding, L. Hu, and G. Xu, "A Heuristic Clustering-Based Task Deployment Approach for Load Balancing Using Bayes Theorem in Cloud Environment," IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 2, pp. 305–316, Feb. 2016, doi: 10.1109/tpds.2015.2402655.

[46] S. Kapoor and C. Dabas, "Cluster based load balancing in cloud computing," 2015 Eighth International Conference on Contemporary Computing (IC3), Aug. 2015, doi: 10.1109/ic3.2015.7346656.

[47] S. Parida and Bakul Panchal, "An Efficient Dynamic Load Balancing Algorithm Using Machine Learning Technique in Cloud Environment," International journal of scientific research in science, engineering and technology, vol. 4, pp. 1184–1186, 2018.

[48] M. Li et al., "Distributed machine learning load balancing strategy in cloud computing services," Wireless Networks, Jul. 2019, doi: 10.1007/s11276-019-02042-2.

[49] X. Sui, D. Liu, L. Li, H. Wang, and H. Yang, "Virtual machine scheduling strategy based on machine learning algorithms for load balancing," EURASIP Journal on Wireless Communications and Networking, vol. 2019, no. 1, Jun. 2019, doi: 10.1186/s13638-019-1454-9.

[50] B. Kruekaew and W. Kimpan, "Multi-Objective Task Scheduling Optimization for Load Balancing in Cloud Computing Environment Using Hybrid Artificial Bee Colony Algorithm With Reinforcement Learning," IEEE Access, vol. 10, pp. 17803–17818, 2022, doi: 10.1109/access.2022.3149955.

[51] D. Rathod and G. Chowdhary, "Load Balancing of Fog Computing Centers: Minimizing Response Time of High Priority Requests," International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 11, pp. 2713–2716, Sep. 2019, doi: 10.35940/ijitee.k2171.0981119.

[52] S. Sthapit, J. R. Hopgood, and J. Thompson, "Distributed computational load balancing for real-time applications," 2017 25th European Signal Processing Conference (EUSIPCO), Aug. 2017, doi: 10.23919/eusipco.2017.8081436.

[53] E. C. Pinto Neto, G. Callou, and F. Aires, "An algorithm to optimise the load distribution of fog environments," 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Oct. 2017, doi: 10.1109/smc.2017.8122791.

[54] S. Ningning, G. Chao, A. Xingshuo, and Z. Qiang, "Fog computing dynamic load balancing mechanism based on graph repartitioning," China Communications, vol. 13, no. 3, pp. 156–164, Mar. 2016, doi: 10.1109/cc.2016.7445510.

[55] J. Jijin, B.-C. Seet, P. H. J. Chong, and H. Jarrah, "Service load balancing in fog-based 5G radio access networks," 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Oct. 2017, doi: 10.1109/pimrc.2017.8292300.

[56] H. A. Khattak et al., "Utilization and load balancing in fog servers for health applications," EURASIP Journal on

Wireless Communications and Networking, vol. 2019, no. 1, Apr. 2019, doi: 10.1186/s13638-019-1395-3.

[57] S. Hamrioui and P. Lorenz, “Load Balancing Algorithm for Efficient and Reliable IoT Communications within E-Health Environment,” GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Dec. 2017, doi: 10.1109/glocom.2017.8254435.

[58] J. L. Crespo-Mariño and E. Meneses-Rojas, Eds., High Performance Computing. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-41005-6.

[59] X. Xu et al., “Dynamic Resource Allocation for Load Balancing in Fog Environment,” Wireless Communications and Mobile Computing, vol. 2018, pp. 1–15, 2018, doi: 10.1155/2018/6421607.

[60] 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, Jul. 2021, doi: 10.1007/s13204-021-01970-w..

[61] R. Beraldi, C. Canali, R. Lancellotti, and G. P. Mattia, “A Random Walk based Load Balancing Algorithm for Fog Computing,” 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), Apr. 2020, doi: 10.1109/fmec49853.2020.9144962.

[62] E. Batista, G. Figueiredo, M. Peixoto, M. Serrano, and C. Prazeres, “Load Balancing in the Fog of Things Platforms Through Software-Defined Networking,” 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Jul. 2018, doi: 10.1109/cybermatics_2018.2018.00297.

[63] R. Beraldi and H. Alnuweiri, “Exploiting Power-of-Choices for Load Balancing in Fog Computing,” 2019 IEEE International Conference on Fog Computing (ICFC), Jun. 2019, doi: 10.1109/icfc.2019.00019.

[64] V. R. Chandakanna and V. K. Vatsavayi, “A sliding window based Self-Learning and Adaptive Load Balancer,” Journal of Network and Computer Applications, vol. 56, pp. 188–205, Oct. 2015, doi: 10.1016/j.jnca.2015.07.001.

[65] Y. Liu, C. Zhang, B. Li, and J. Niu, “DeMS: A hybrid scheme of task scheduling and load balancing in computing clusters,” Journal of Network and Computer Applications, vol. 83, pp. 213–220, Apr. 2017, doi: 10.1016/j.jnca.2015.04.017.

[66] S. S. Patil and A. N. Gopal, "Dynamic Load Balancing Using Periodically Load Collection with Past Experience Policy on Linux Cluster System," American Journal of Mathematical and Computer Modelling, vol. 2, No. 2, pp. 60–75, 2017, doi: 10.11648/j.ajmcm.20170202.13.

[67] V. Priya, C. Sathiya Kumar, and R. Kannan, “Resource scheduling algorithm with load balancing for cloud service provisioning,” Applied Soft Computing, vol. 76, pp. 416–424, Mar. 2019, doi: 10.1016/j.asoc.2018.12.021.

[68] D. F. Altayeb and F. A. Mustafa, "Analysis on Load Balancing Algorithms Implementation on Cloud Computing," International Journal of Innovative Research in Advanced Engineering, vol. 6, no. 2, pp. 1-32, 2016.

[69] S. L. Chen, Y. Y. Chen, and S. H. Kuo, “CLB: A novel load balancing architecture and algorithm for cloud services,” Computers & Electrical Engineering, vol. 58, pp. 154–160, Feb. 2017, doi: 10.1016/j.compeleceng.2016.01.029.

[70] S. L. Mirtaheri, L. Grandinetti, "Optimized Dynamic Load Balancing in Distributed Exascale Computing Systems," Ph.D. Thesis, Dept. of Electronics, Informatics and Systems Engineering, Univ. of Calabria, Italy, 2016, doi: 10.13126/unical.it/dottorati/1370.

[71] A. Bhandari and K. Kaur, “An Enhanced Post-migration Algorithm for Dynamic Load Balancing in Cloud Computing Environment,” Advances in Intelligent Systems and Computing, pp. 59–73, Oct. 2018, doi: 10.1007/978-981-13-1544-2_6.

[72] H. Naseri, S. Azizi, and A. Abdollahpouri, "BSFS: A Bidirectional Search Algorithm for Flow Scheduling in Cloud Data Centers," Journal of Information Systems and Telecommunication (JIST), vol. 3, no. 27, p. 175, Mar. 2020, doi: https://doi.org/10.7508/jist.2019.03.002.

[73] M. Kaur and R. Aron, “Energy-aware load balancing in fog cloud computing,” Materials Today: Proceedings, Dec. 2020, doi: 10.1016/j.matpr.2020.11.121.

[74] R. O. Aburukba, M. AliKarrar, T. Landolsi, and K. El-Fakih, “Scheduling Internet of Things requests to minimize latency in hybrid Fog–Cloud computing,” Future Generation Computer Systems, vol. 111, pp. 539–551, Oct. 2020, doi: 10.1016/j.future.2019.09.039.

[75] J. Bisht and V. Subrahmanyam, “Survey on Load Balancing and Scheduling Algorithms in Cloud Integrated Fog Environment,” Proceedings of the 2nd International Conference on ICT for Digital, Smart, and Sustainable Development, ICIDSSD 2020, 27-28 February 2020, Jamia Hamdard, New Delhi, India, 2021, doi: 10.4108/eai.27-2-2020.2303123.

[76] A. Alarifi, F. Abdelsamie, and M. Amoon, “A fault-tolerant aware scheduling method for fog-cloud environments,” PLOS ONE, vol. 14, no. 10, p. e0223902, Oct. 2019, doi: 10.1371/journal.pone.0223902.

[77] F. Alqahtani, M. Amoon, and A. A. Nasr, “Reliable scheduling and load balancing for requests in cloud-fog computing,” Peer-to-Peer Networking and Applications, vol. 14, no. 4, pp. 1905–1916, Mar. 2021, doi: 10.1007/s12083-021-01125-2.

[78] F. Fakhar, "Investigate Network Simulation Tools in Designing and Managing Intelligent Systems," Journal of Information Systems and Telecommunication (JIST), vol. 28, no. 7, pp. 278-293, Jun. 2020, doi: 10.7508/jist.2019.04.004.

[79] E. M. Khaneghah, S. L. Mirtaheri and M. Sharifi, "Evaluating the Effect of Inter Process Communication Efficiency on High Performance Distributed Scientific Computing," 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, Shanghai, China, 2008, pp. 366-372, doi: 10.1109/EUC.2008.11.