IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 30, 2020, accepted July 4, 2020, date of publication July 14, 2020, date of current version July 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009184

A Comprehensive Study of Load Balancing Approaches in the Cloud Computing Environment and a Novel Fault Tolerance Approach

MUHAMMAD ASIM SHAHID[1,2], NOMAN ISLAM[1,3], MUHAMMAD MANSOOR ALAM[1,4], MAZLIHAM MOHD SU'UD[1], AND SHAHRULNIZA MUSA[1]

[1]Malaysian Institute of Information Technology, Universiti Kuala Lumpur, Kuala Lumpur 50250, Malaysia
[2]Computer Science Department, Sir Syed University of Engineering & Technology, Karachi 75190, Pakistan
[3]Computer Science Department, Iqra University, Karachi 75500, Pakistan
[4]College of CS and Information System, Karachi 75190, Pakistan

Corresponding author: Shahrulniza Musa (shahrulniza@unikl.edu.my)

ABSTRACT The past few years have witnessed the emergence of a novel paradigm called cloud computing. CC aims to provide computation and resources over the internet via dynamic provisioning of services. There are several challenges and issues associated with implementation of CC. This research paper deliberates on one of CC main problems i.e. load balancing (LB). The goal of LB is equilibrating the computation on the cloud servers such that no host is under/ overloaded. Several LB algorithms have been implemented in literature to provide effective administration and satisfying customer requests for appropriate cloud nodes, to improve the overall efficiency of cloud services, and to provide the end user with more satisfaction. An efficient LB algorithm improves efficiency and asset's usage through effectively spreading the workload across the system's different nodes. This review research paper objective is to present critical study of existing techniques of LB, to discuss various LB parameters i.e. throughput, performance, migration time, response time, overhead, resource usage, scalability, fault tolerance, power savings, etc. The research paper also discusses the problems of LB in the CC environment and identifies the need for a novel LB algorithm that employs FT metrics. It has been found that traditional LB algorithms are not good enough and they do not consider FT efficiency metrics for their operation. Hence, the research paper identifies the need for FT efficiency metric in LB algorithms which is one of the main concerns in cloud environments. A novel algorithm that employs FT in LB is therefore proposed.

INDEX TERMS Cloud computing, load balancing techniques, fault tolerance, load balancing metrics.

I. INTRODUCTION

Cloud computing has emerged as a novel trend in past few years. It has led to the progression of distributed system to a large scale computing network. CC firms such as IBM, Amazon, Yahoo & Google deliver cloud services for consumers around the globe. In this novel paradigm, end users are not required to install apps into their local computers; instead apps and services are offered on-demand to end-users [1]. There have been various challenges in true realization of a cloud environment. Among those challenges, LB is an issue of prime concern. It is defined as how the load is shared between different machines [2]. LB implies the distribution of the necessary burden across various computer solutions, computer clusters, such as servers, links to the network, disks, CPUs, etc. [3]. LB offers approaches to maximize the system output, resource usage and device performance. It offers us one of the benefits of keeping data or files in an easy and scalable way and makes them accessible to customers on a large scale. There are many LB algorithms in the cloud system to make the most effective use of resources [4].

This research paper discusses major challenges of CC and then deliberates on LB problem. The major goals in this research paper are as follows:

(a) To identify major challenges in CC
(b) To review different approaches proposed for LB
(c) To identify the need for a novel LB policy that employs FT metrics

The associate editor coordinating the review of this manuscript and approving it for publication was Sabu M. Thampi.

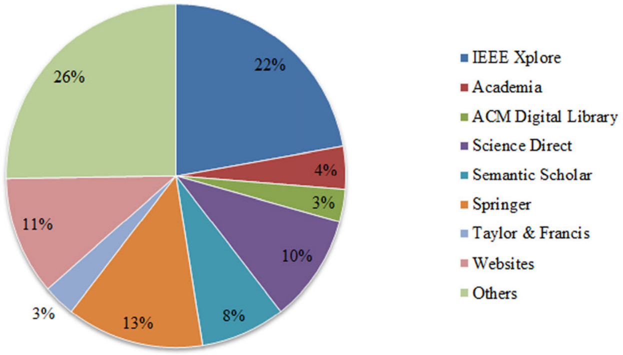


FIGURE 1. The percentage of research paper read from multiple sources.

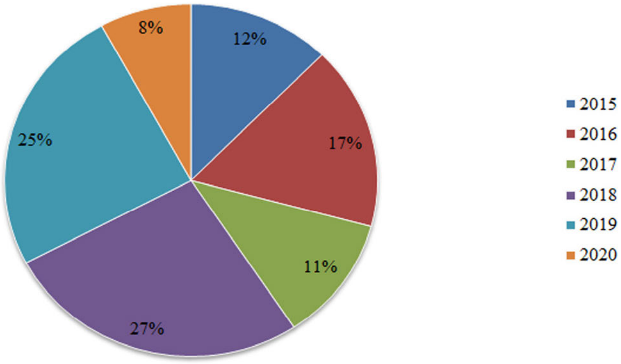


FIGURE 2. The percentage of paper read between 2015 to 2020.

(d) To propose a novel LB algorithm based on FT metrics

So, the research paper primarily deliberates on LB issues in cloud computing environment.

A. MOTIVATION & THE NEED FOR LOAD BALANCING

The objective of LB is to spread the burden onto VMS in equal proportion for optimal use of the resource. Diverse LB algorithms are explored in this survey depending on various parameters [7].

What is meant by load balancer?

The main aim of the load balancer helps to assign resources equally to the tasks for resource efficiency and user satisfaction at minimal expense [7], quality output, gripping rapid traffic blast sustain traffic on the website and elasticity which motivates us to identify problems in LB and to work on their resolution [105]. This plays a key role in ensuring the ease of access for customers, business partners, and end-users of the cloud-based applications [104]. The potential of the load balancing and its various applications provide motivation to deliberate on this important challenge, to identify major issues in LB and to resolve these issues [7].

The review research paper is set out as follows: Section **2** provides the context and design of this research. Section **3** discusses in detail the CC challenges and issues. Section **4** discusses in detail LB challenges, parameters, policies, and classification of process state-based LB approaches in the cloud. Section **5** Machine learning for load balancing. Section **6** describes the study of present LB methods. Section **7** Fault tolerance and taxonomy of FT. Section **8** contains the research synthesis of various LB methods, a comparison of different LB methods, based on various parameters, and the significance of FT. Section **9** discusses the proposed work that resolves the issues of the various conventional LB algorithms via proposing an efficient FT LB technique. The proposed technique is a resilient /adaptive method that works with the help of ML and AI. Section **10** presents the future directions of LB in the cloud environment. Section **11**presents concluding remarks.

TABLE 1. A detailed description of various load balancers.

Load Balancer	Explanation
Hardware load balancer	HLD is a physical unit that maintains the specific server in a network and is used to distribute web traffic to different servers [76].
Network load balancer	NLB operates in the OSI layout network layer or Layer 4. This is suitable for LB TCP traffic [76].
Application load balancer	ALB serves at OSI platform layer 7. For high-level loads it is total Equilibrations of HTTP and HTTPS [76].
Classic load balancer	CLB provides simple LB over different Elastic Cloud Compute instances [76].
Elastic load balancer	It is also identified as an AWS balance load balancer. It hands out incoming tasks in several cases on Amazon EC2 [76].
HA proxy load balancer	Its setup has 2 elements: one for users and one for users wards LAN Server [76].

II. CONTEXT AND DESIGN OF RESEARCH

The next few paragraphs describe the research design and discuss the set of research papers explored in research along with data sources and exploration criteria.

A. RESEARCH QUESTIONS

The main research questions that are addressed in this research are as follows:

(a) What are the current state-of-the-art challenges in CC?
(b) What is the significance of LB cloud computing?
(c) What are the various LB techniques currently available in literature and can taxonomy is developed for those techniques?
(d) What are the current performance parameters and the manner in which these have been applied to LB?
(e) What are the research gap existing in current LB techniques?
(f) Can a new LB algorithm be developed that can address the gaps identified?
(g) What is the future of CC? What challenges exist?

The above questions have been answered by presenting effective and accurate CC and LB information based on the research paper, under the research path. The responses are described below:

(a) What are the current state-of-the-art challenges in cloud computing?

In the light of the research paper, the challenges associated with CC have many. Figure 3 shows the taxonomy of the current state of the art challenges associated with CC. The cloud computing challenges has: data protection, data recovery and availability, administrative capabilities, regulation and compliance restrictions [5], security, capable of adjusting the burden, controlling executions [6], load balancing, fault tolerance [57], cloud computing governance [100], interoperability and portability [101].

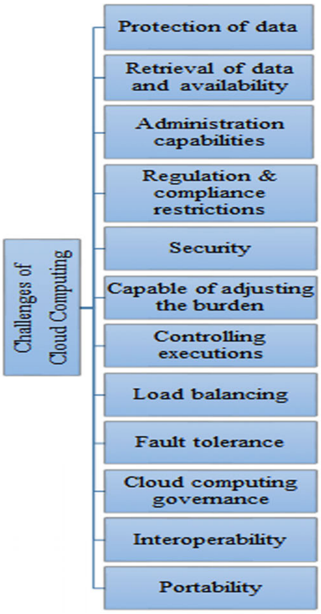

Challenges of Cloud Computing
Protection of data
Retrieval of data and availability
Administration capabilities
Regulation & compliance restrictions
Security
Capable of adjusting the burden
Controlling executions
Load balancing
Fault tolerance
Cloud computing governance
Interoperability
Portability

FIGURE 3. Taxonomy of major problems in CC [5].

(b) What is the significance of load balancing cloud computing?

This query is aimed at recognizing the importance of LB in cloud computing. LB is an essential part of any cloud environment. It plays a vital role in keeping the ease of access for customers, business partners, and end-users of your cloud-based applications. LB is greatly advantageous for cloud environments, where tremendous workloads could rapidly overwhelm a single server, rising availability of service and response times are crucial to some business operations or are permitted by SLAs. Without LB, newly spinning virtual servers will be unable or at all to accept the incoming traffic in a coordinated fashion. Few virtual servers may also be left to handle zero traffic while others may have been overwhelmed. Load balancing is also able to identify unobtainable servers and redirect traffic to those that are still in operating condition [104].

(c) What are the various LB techniques currently available in literature and can taxonomy is developed for those techniques?

This research paper shows that the load balancing techniques are combined into four namely geographical distributions, general LB, natural phenomena-based LB & Network-aware task scheduling LB. The geographical distributions of nodes are significant, particularly for large-scale apps such as twitter, Facebook, and so on. Geographic distribution could be defined as a set of decisions on the digital deployment and/or relocation of VMs or computing activities to geographically dispersed data centers in order to reach SLAs or system deadlines for virtual machines/activities and to reduce operating costs for cloud systems [11]. The general LB has some kinds of LB techniques such as round robin, randomized algorithm, threshold algorithm, OLB, OLB + LBMM, min-min, max-min, equally spread current execution algorithm, central LB Strategy for VMs, throttled LB, stochastic hill climbing, and join idle queue [22]. This technique is quick and efficient, but typically the connected servers cannot be found, this leads to inconsistent resource distribution. The biggest problem with this kind of approach is that the actual state of the system is given little consideration to decision making [9]. The natural phenomena-based LB has some kinds of LB techniques such as ant colony algorithm, genetic algorithm, honey bee foraging, artificial bee colony algorithm, hybrid (Ant colony, a honey bee with dynamic feedback), ant colony & complex network LB, osmosis LB algorithm, bee colony optimization algorithm, and LB honey bee foraging. This technique influenced by natural phenomena or biological behavior [21]. The network-aware task scheduling LB has some types of LB techniques like shortest job scheduling LB algorithm, task scheduling strategy based on LB, active clustering and biased random sampling [22].

'Yes'. The taxonomy may be proposed for those techniques. The techniques of LB can be divided into four specific categories namely geographical distributions, general LB, natural phenomena-based LB & Network-aware task scheduling LB based on different sub types of LB techniques [12].

(d) What are the current performance parameters and the manner in which these have been applied to LB?

In CC analysis it is helpful to determine different parameters that will help verify LB techniques. The current LB performance parameters based on four distinct groups namely (I) LB performance parameter with qualitative attributes & dependent nature. (II) LB performance parameters with qualitative attributes & independent nature (III) LB performance parameters with quantitative attributes & dependent nature (IV) LB performance parameters with quantitative attributes & independent nature. The LB performance parameter with qualitative attributes & dependent nature has an overhead performance parameter. The LB performance parameters with qualitative attributes & independent nature have some kinds of performance parameters such as scalability & fault tolerance. The LB performance parameters with quantitative attributes & dependent nature have some types of performance parameters such as throughput, migration time, resource utilization factor & power saving.

The LB performance parameters with quantitative attributes & independent nature have some kinds of performance parameters such as response time & performance [10].

The performance parameters have been applied to LB such as performance, throughput, overhead, fault tolerance, migration time, response time, resource utilization, scalability and power saving [10]. The parameters concerning cloud LB in a much more practical sense will not only enhance output processing by LB the process but also make the theoretical basis for studying efficient algorithms to boost LB efficiency on CC [103].

(e) What are the research gap existing in current load balancing techniques?

The research paper identified limitations and remaining issues with existing and newly researched and implemented load balancing techniques. Could be listed as follows:

- Enhanced LB performance
- Having an ensuring the continuity, even if incompletely the system fails.
- Device consistency must be preserved at regular intervals.
- Services are accessible immediately upon request.
- The various traditional load balancing techniques do not perform well and do not work based on fault tolerance performance metrics.

(f) Can a new load balancing algorithm be developed that can address the gaps identified?

'Yes'. The extensively analyzed the relevant research paper and defined the load balancing in cloud computing most widely recognized. This research paper, proposes an efficient fault tolerance LB technique that ensures will properly address the research gaps. This technique combines resilient methods/adaptive methods such as machine learning and artificial intelligence have played an active role in the RSM domain. The RSMs, tend to be the potential path of research gaps. By current definition, a system's resilience is the indicator of how easily and the system can recover quickly and continue to function properly after a system outage or failure has occurred. The capacity of RSMs to assure a customer reaction is directly relevant to system consistency that forms part of Service quality (QOS) [68]. The proposed LB technique will has various properties as follows:

- **Adaptability:** Systems must have been able to adapt to their environment [69].
- **Throughput:** Any system must be in place to maintain the number of tasks performed each unit time [69].
- **Scalable:** The number of extra resources the machine requires to compensate from the fault [69].
- **Response time:** This will preferably be on its lowest value [69].
- **Usability:** The number of resources that the user requires to accomplish a given task will be at its lowest rate [69].
- **Availability:** Another significant consideration is the no of times a tool is open to users at the time [69].
- **Associated overheads:** The use of extra resources required by the process of fault tolerance to retrieve the device from fault is addressed [69].

(g) What is the future of cloud computing? What challenges exist?

This query is aimed at recognizing the future of CC. Cloud computing future as a mix of cloud-based tech tools and on-site computing to help develop hybrid IT solutions. Cloud has several functions, making it easier for the IT sector in the future. Here are several CC factors or forecasts as follows [108]:

- **Strengthen Space Capacities:** Data are producing at a high volume today and it's hard to safely store it [108].
- **Network Performance Improved**: The efficiency of the internet can be improved with the help of the Internet of Things [108].
- **Customizable Applications go forward:** Including the complexity, the volume of a single program is rising constantly [108].
- **IoT inside Cloud Technology:** The IoT is also among the leading technologies that arrive with constant progress in Data Analytics and Cloud computing in real-time [108].
- **Information also reveals how Future Changes:** The demand for CC is rising at 22.8 percent and after 2018 will reach $127.5. 62 percent of all CRM apps will be cloud-based by 2018. In comparison, software as a service-based technology accounts for 30 percent of all technology spending [108].
- **Greater Cloud hosting:** CC is customer-friendly and compliant with both older and newer organizations [108].
- **Secure:** The data that is contained in the cloud is safe yet not complete. The small businesses that can provide cloud services might or might not provide the data with sufficient protection [108].
- **Customizable Applications:** Companies use a lot of software, which has yet to be updated. This leads to CC needing updated software, offering better protection and services [108].
- **Financial:** As cloud infrastructure continues to grow, the use of the hardware would be limited such as the bulk of the work would be performed using CC and virtualization [108].
- **Server less Computation:** Standard cloud computing involves the running of an app on a VM which in effect provides a service to the customer [109].
- **Elasticity & Scalability:** Research directions in scalability and elasticity for the next decade may be broken down into equipment, content management, and application-level [110].

Several challenges exists can be taken into the future of CC. Those challenges are shown below:

- **Edge of Computing:** Edge computing is a novel concept that allows the processing of the data generated by the IoT [110].

- **Boost Difficulty:** Nowadays, peoples are much more familiar with technical difficulties than it is with IT production [110].
- **Crypto currency:** Crypto currency is electronic money; it's produced using a sort of technology [110].
- **Internet of Thing:** IoT is one of the most significant developments and is slowly growing [110].

B. SERACH CRITERIA

A systematic analysis of LB and CC were performed over well-known research. Following search string words were used: cloud computing, load balancing challenges and issues in LB and CC. It also includes LB performance parameter with qualitative and quantitative attributes based on dependent and independent nature, policies of LB, different types of LB methods, fault tolerance, etc. In this research, the search engine list is mentioned in Table 2.

TABLE 2. Finding engine choose.

Finding Engine	Source Address
IEEE Xplore	http://ieeexplore.ieee.org
Academia	http://academia.edu
ACM Digital Library	http://acm.org
Science Direct	http://sciencedirect.com)
Semantic Scholar	http://semanticscholar.org
Springer	http://springer.com
Taylor & Francis	https://www.taylorandfrancis.com

C. DATA SOURCES

For this survey, a number of diverse data sources were considered. The research paper widely looked for conferences and journal research papers in Scopus, blogs, Google Scholar, books, and magazines as a database for the extraction of related research papers. In our quest the following databases were used:

D. EXPLORATION CRITERIA

The research was performed from 2015 to January 2020. From the research papers scanned include the research papers that meet the quality evaluation. This covers research papers from peer-reviewed Scopus, Journals, Google Scholar, books, blogs, and white papers. Figure 2, displays the percentage of papers read between 2015 to 2020.

E. QUALITY ASSESMENT

The context of the research is one of the potential ways in which various forms of quantitative specific research are carried out. On the searched research papers, quality assessment criteria were applied for inclusion and exclusion of research papers. After initial study of abstract, some of the research papers were excluded. Then the research papers were studied completely and based on criteria, the research papers were either excluded or included for review. The major inclusion criteria for research papers are:

(a) The major focus on the LB algorithms for CC.
(b) Secondly is to select those research papers that define the challenges of LB, CC, and the different LB efficiency metrics based on load balance algorithms.

TABLE 3. Inclusion and exclusion criteria.

	Criteria
Inclusion	• A research paper that outlined clearly how the load balancing described in the CC environment techniques can be adjusted and assisted. • A research paper that is developed by either academics or practitioners. • A research paper conducted in the context of cloud computing. • A research paper conducted in the context of load balancing. • A research paper is peer-reviewed. • A research paper that is written in English.
Exclusion	• A research paper which contains only journal papers. • A research paper whose emphasis is not on LB techniques in CC environments.

Before proceeding towards discussion on load balancing, let's first explore some of the major challenges in cloud computing that must be dealt with for realizing cloud computing real potential [5], [6].

III. THE CHALLENGES OF CLOUD COMPUTING

There has been a lot of challenges associated with CC. Figure 3 demonstrates the taxonomy of major problems with the CC [5]. This includes: data protection [5], data recovery and availability [5], administrative capabilities [5], regulation and compliance restrictions [5], security [6], capable of adjusting the burden [6], controlling executions [6], load balance, fault tolerance [57], cloud computing governance [100], interoperability and portability [101].

Let's speak in-depth about each of those things:

1) **Protection of data:** Data protection is a key aspect that needs to be taken into account. By putting the data on the cloud, the issue of privacy still arises. Similarly, in many situations the precise placement of repository sites is never known, adding to the organization's privacy concerns. In most existing models, the knowledge is protected by firewalls via datacenters (owned by the company) [5].
2) **Data retrieval and availability issue:** It should be remembered that the Service Level Agreements (SLAs) will comply entirely with the company requirements. The operational staffs here have a big role to play in overseeing service level agreements and running system

time management. In addition, there is a role that involves supporting management units [5]:

(a) Data Replication [5]
(b) Adequate clustering & failure [5]
(c) Device control (tracking transactions, log tracking) [5]
(d) Disaster recovery [5]
(e) Managing Power & Efficiency [5]
(f) Maintenance (Runtime Governance) [5]

3) **Administrative capabilities:** Although there are plenty of cloud services available, the initial step is infrastructure management, network transformation Dynamic-scaling functionality, dynamic-resource in several organizations, allocation, for example, is a key necessity. There is tremendous potential for enhancing the robustness and LB functionality that has been given to date [5].

4) **Regulation and compliance restrictions:** In several European countries, policy regulations do not mandate user private information and other confidential information to be stored physically and outside of the nation or state. Cloud service providers also need to establish includes a data hub or storage area mainly inside the country to comply with regulations to meet such criteria. It may not always be viable to have such an infrastructure and is a big challenge for cloud providers [5].

5) **Security:** CC security or, more commonly, cloud security related to a wide range of legislation, technology, applications and controls it uses to protect the virtualized Internet, data, applications, services and associated CC infrastructure. It's an information security sub-domain, internet security, and, broader, computer security. Also, digital identities, username & password must be guarded in the cloud as should any data collected or produced by the supplier regarding customer action [58].

6) **Capable of adjusting the burden:** The tension adjustment passes through all the cores of each stack. This also increases device output. Numerous current figures offer a modification to the stack and efficient use of resources. There are several possibilities for creating stack in the cloud such as memory, CPU, and structure stack. Changing the strain is the road to having center point's overload and then moving the additional store to other centers [6].

7) **Execution monitoring:** Cloud computing requires several different levels of software which are given as utilities. Such is Web Infrastructure, Data Network, & Applications. Execution control has three levels: Infrastructure, Device & Application levels. The first two components are used to set up the network and gather data from different agencies that are spread around the cloud. The third component is the part of data processing used to set up and to cause items based on certain circumstances [59].

8) **Load balancing:** Load balancing is a major issue with the CC now, stopping some nodes from being overwhelmed while others are idle and have to do some work. LB may improve the QoS metrics including cost, response time, reliability, efficiency, and resource utilization [22]. The nature of LB operations, the complexity of the constructed algorithm, should be as minimal as possible in order to avoid errors and wait in complicated operations [96].

9) **Fault tolerance:** FT is one of the most critical parameters due to resource dropping impacts on unit performance, job outcomes, productivity, response time, & quality. Therefore, a fault tolerance strategy is required for recognizing faults, correcting those faults, and thus boosting performance parameters. Fault tolerance is a key concern to ensure the consistency of the core services and the completion of the program [60].

10) **Cloud computing governance:** Cloud computing has experienced wide acceptance and use in the past and the current decade. Regardless of the importance of cloud computing in enhancing organizational performance, its governance plays a role. A critical function in decision making. Cloud computing governance can be regarded as part of the general IT-management umbrella [100].

11) **Interoperability:** The single platform program should be able to integrate resources from the other platform. This is called Interoperability. Via web services and it's becoming conceivable but designing these web services is complex [101].

12) **Portability:** Apps working on one cloud platform could be moved into a new cloud platform, and it should function properly without making any layout, programming alterations. Portability isn't achievable because different standard languages are used by each cloud provider for their framework. [101].

IV. LOAD BALANCING

LB's primary objective is to efficiently manage the load across various cloud nodes, so that no node is under/ overloaded [7]. LB may be characterized as a process of spreading a burden across network links on multiple devices or system clusters to maximize its use of assets to optimize overall response time. It reduces the device's total waiting period and also avoids excessive replication of assets. Requests spread inside servers in this process so that data can be distributed & processed without waiting. LB is the method of maximizing system performance by moving the device burden [9]. The LB at CC is shown in Figure 4.

LB provides a systematic mechanism for the equal distribution of the responsibility to the resources available. The goal is to provide reliable service, including adequate use of the resource, in the event of a disaster of the portion of any service by supplying & de-provisioning the device instance. In addition, LB is aimed at reducing response time for tasks & increasing resource efficiency, which increases device efficiency at a lower cost [9].

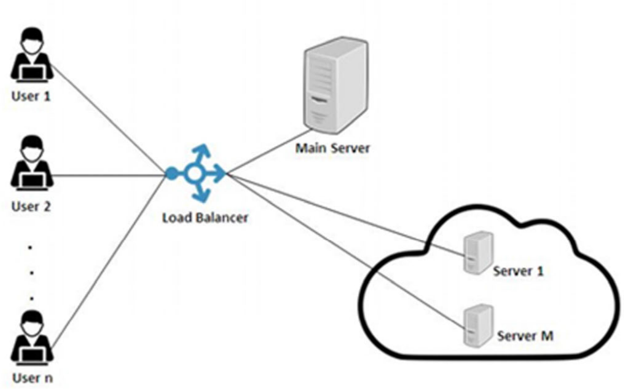


FIGURE 4. Load balancing in Cloud Computing [8].

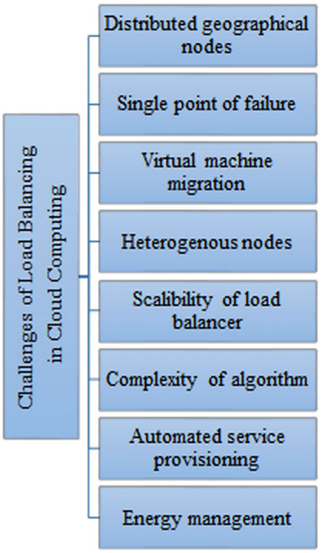

Challenges of Load Balancing in Cloud Computing
Distributed geographical nodes
Single point of failure
Virtual machine migration
Heterogenous nodes
Scalibility of load balancer
Complexity of algorithm
Automated service provisioning
Energy management

FIGURE 5. Taxonomy of LB challenges in CC [9].

A. CHALLENGES OF LB IN CC

CC faces many challenges; with LB as one of the most critical problems needing specific attention. This includes issues such as (VM) migration, virtual machine security; user QoS comfort & resource use get equal attention to seeking a better solution to improve cloud resource use. Below is a list of a few LB issues and Figure 5 shows the taxonomy of critical LB issues [9]:

1) **Distributed Geographical Nodes:** Cloud data centers are typically distributed for computing at disparate locations. Dynamically distributed nodes in these centers are used as a centralized network for efficient processing of customer requests. Several LB approaches are available in the literature with a limited reach and where conditions such as network delay, communication delay, the range within the distributed computing nodes, space within customer & resources are not taken into consideration. Nodes in very remote areas are challenging because certain algorithms do not suit this environment [9].
2) **Single Point of Failure:** Specific LB algorithms are proposed in literature where decision-making is not distributed across multiple nodes, and LB decisions are made by the centralized node. If the key devices malfunction this will impact the overall computing system [9].
3) **VM Migration:** Virtualization allows for the building of multiple virtual machines on one physical unit. Those virtual machines have different settings & are autonomous in architecture. If a physical device is overloaded, it is appropriate to shift all VMs to a remote location using an LB method to relocate the VM [9].
4) **Nodes Heterogeneity:** The authors have proposed homogeneous nodes in the cloud load balancing in the initial inquiry. CC consumers need a dynamic switch, which needs execution on heterogeneous nodes for an efficient network and reduces response time [9].
5) **Handling Data:** CC addressed the issue of old conventional storage devices which demanded huge resource & equipment costs for hardware. The cloud allows consumers to heterogeneously retain the data, without any control issues. Storage is increasing day by day and requires duplication of stored data for effective accessibility & data continuity [9].
6) **LB Scalability:** Accessibility & on-demand scalability cloud services allow people to access resources for rapid downscaling or scale-up at any time. A strong load balance should consider rapidly changing requirements in computational conditions, memory, device topology, etc. [9].
7) **Complexity of Algorithm:** CC algorithms should be quick & simple to achieve. The aim of a robust algorithm is to reduce cloud system efficiency & quality [9].
8) **Automated Service Provisioning:** The key aspect associated with cloud computing is flexibility; resources can be automatically delegated or distributed. How then do we use or discharge the cloud's services, only maintaining the same productivity as conventional systems and using the best resource [102].
9) **Energy Management:** The benefits of energy management, which advocates cloud use, are the economies of scale. Power saving is the most important thing that allows for a global economy where limited companies are going to help the pool of worldwide capital, rather than each providing its private services [102].

B. PARAMETERS OF LB

The parameters concerning cloud LB in a much more practical sense will not only enhance output processing by LB the process but also make the theoretical basis for studying efficient algorithms to boost LB efficiency on CC [103].

LB refers to the efficient methods used for cloud workload allocation between VMs. Within a cloud network, the versatility of the VMs depends on the degree of load distributed across existing resources. A decent scheduler allows for a reliable method of load control. Parameters of CC, namely,

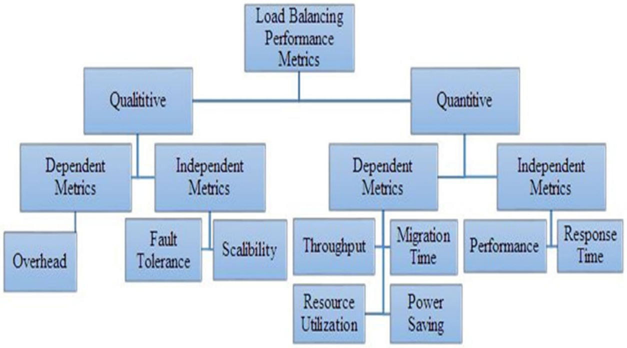


FIGURE 6. Taxonomy of LB parameters [10].

are important. The performance measurements recognized in the LB methods are divided into two major quantitative & qualitative parameter classifications. In fact, the parameters can also be either receptive or autonomous. The taxonomy of LB metrics is shown in Figure 6 [10].

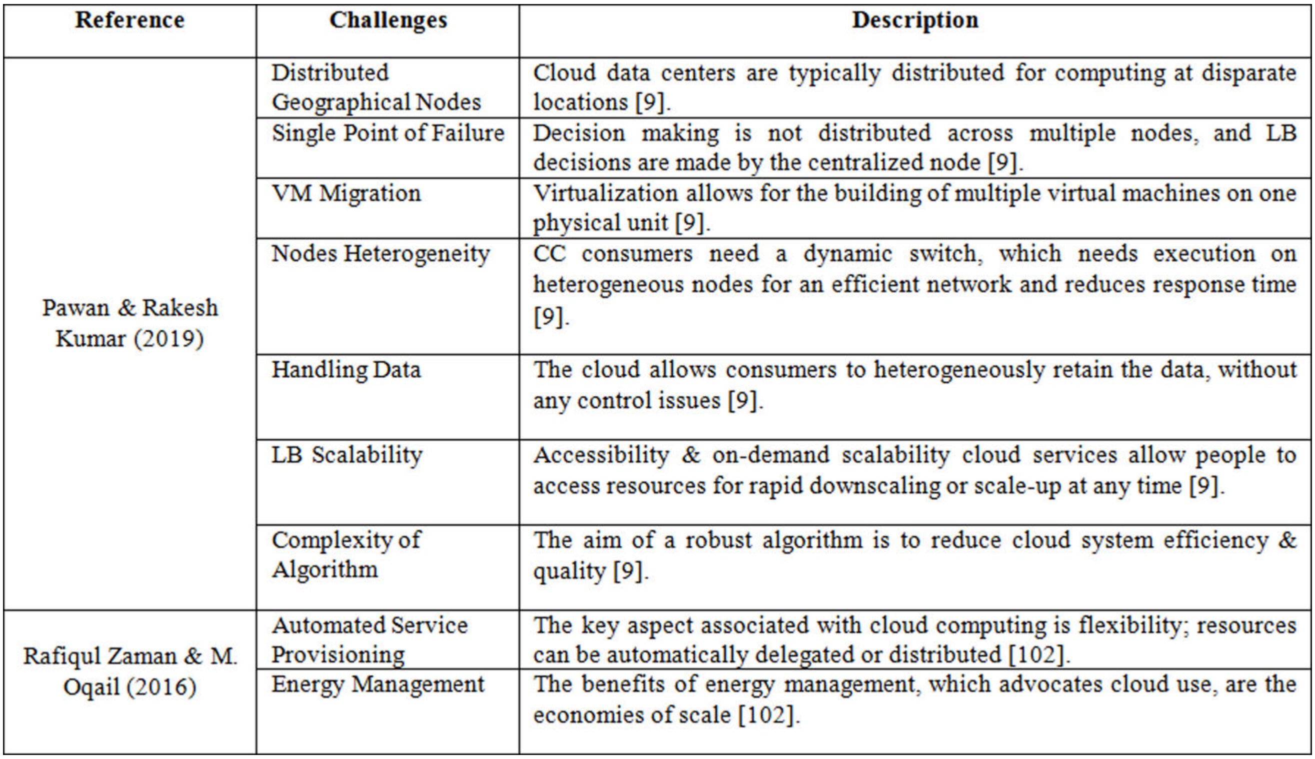
In addition to the current load balance parameters, there are a few performance parameters added in this work. Furthermore, if novel parameters are identified in the future, they may be put according to their characteristics in the categorization. Taxonomy classifies the cloud LB parameters into four distinct groups [10]:

1) LB performance parameter with qualitative attributes & dependent nature [10].
2) LB performance parameters with qualitative attributes & independent nature [10].
3) LB performance parameters with quantitative attributes & dependent nature [10].
4) LB performance parameters with quantitative attributes & independent nature [10].

The foregoing parameters are combined behind a common category called service quality metrics (QoS) [10].

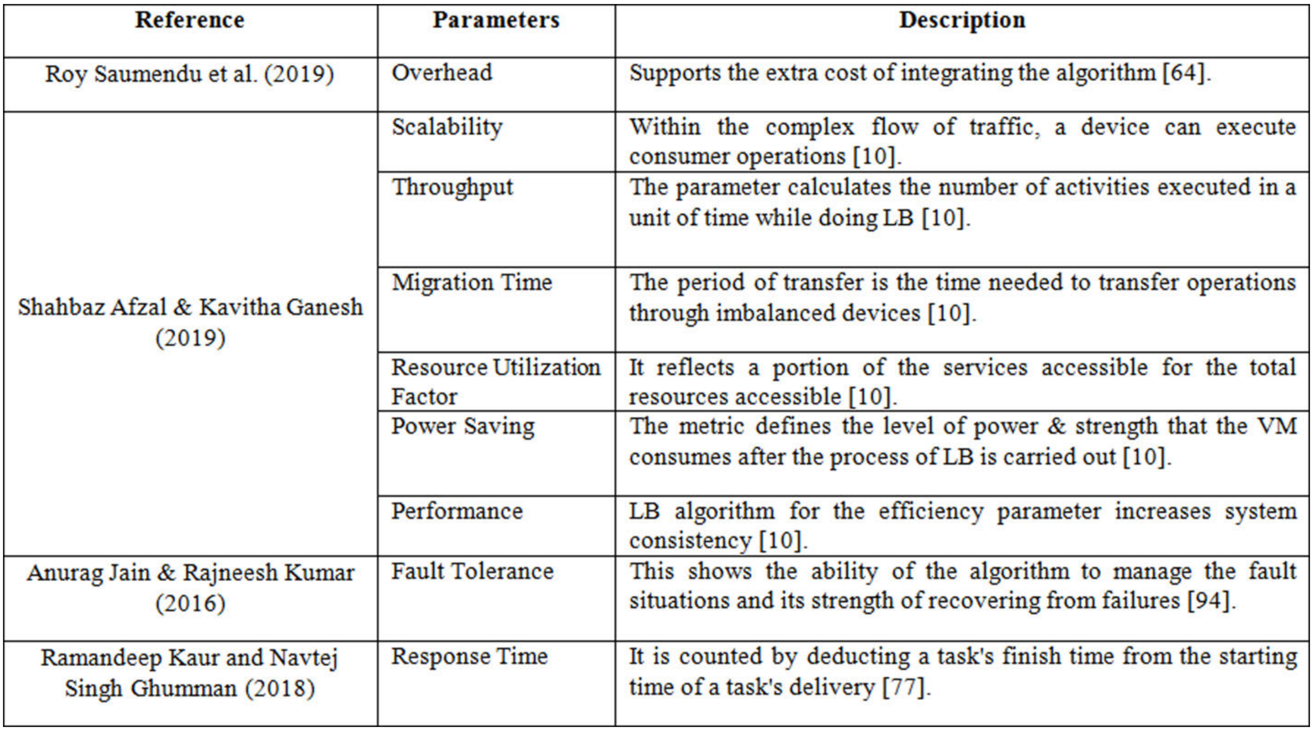
1) LB performance parameter with qualitative attributes & dependent nature.
   The following parameters are used in the LB method [10]:
   (a) **Overhead:** The overhead associated with any LB algorithm supports the extra cost of integrating the algorithm [64].
2) LB performance parameters with qualitative attributes & independent nature.
   This kind of LB metrics is dealt with as follows [10]:
   (b) **Scalability:** Within the complex flow of traffic, a device can execute consumer operations. Respectively, the LB algorithm ought to be able to increase resources in peak periods, & downscale in off-peak times [10]. This indicates the survival rate for a functioning program, whether the amount or volume of the job or workload is raised [76]. The numbers of a node in a process have no impact on the algorithm's fault tolerance power [95].
   (b) **Fault Tolerance:** The capability of a system to operate consistently during any moment of system failure which ultimately results in improved robustness & availability. A fault-tolerant LB algorithm would guarantee minimum network loss due to network overload or other [10]. This shows the ability of the algorithm to manage the fault situations and its strength of recovering from failures [94].
3) LB performance parameters with quantitative attributes & dependent nature.
   The results parameter which can be measured and which depending on certain variables in each form or another is described as follows [10]:
   (a) **Throughput:** The parameter calculates the number of activities executed in a unit of time while doing LB. This defines the level at which computing job is performed using a LB algorithm. The objective of the LB algorithm is to gain greater performance [10]. Tasks that have accomplished their fulfillment inside a specified period [72] and maximum no. of the dead (or served) function each unit of time [93].
   (b) **Migration Time:** The period of transfer is the time needed to transfer operations through imbalanced devices. This could also be the time required to move the overloaded VMs via one Physical Machine (PM) to the next physical machine, like in the virtual machine transfer LB [10].
   (c) **Resource Utilization Factor:** It reflects a portion of the services accessible for the total resources accessible. This determines to what degree a VM uses the tools. If a VM gets overwhelmed, the tasks consume much of the energy, but this is an unwanted phenomenon because the tasks cannot be done quickly. Greater resource use means greater resource usage which suggests depleted resources which in turn means few free services. Therefore an effective LB algorithm makes the best the most of the resources [10].
   (d) **Power Saving:** The metric defines the level of power & strength that the VM consumes after the process of LB is carried out. An effective algorithm for LB decreases power & energy usage in a virtual machine [10].
4) LB performance parameters with quantitative attributes & independent nature.
   Some types of parameters mentioned below are: [10]
   (a) **Response Time:** This is the overall period a device requires to react to a user request & is mathematically equal to the total of time in service & stack wait time while avoiding the transmitting time maintaining the reliability attribute [10]. It is counted by deducting a task's

TABLE 4. The summary of LB challenges in.



Reference	Challenges	Description
Pawan & Rakesh Kumar (2019)	Distributed Geographical Nodes	Cloud data centers are typically distributed for computing at disparate locations [9].
	Single Point of Failure	Decision making is not distributed across multiple nodes, and LB decisions are made by the centralized node [9].
	VM Migration	Virtualization allows for the building of multiple virtual machines on one physical unit [9].
	Nodes Heterogeneity	CC consumers need a dynamic switch, which needs execution on heterogeneous nodes for an efficient network and reduces response time [9].
	Handling Data	The cloud allows consumers to heterogeneously retain the data, without any control issues [9].
	LB Scalability	Accessibility & on-demand scalability cloud services allow people to access resources for rapid downscaling or scale-up at any time [9].
	Complexity of Algorithm	The aim of a robust algorithm is to reduce cloud system efficiency & quality [9].
Rafiqul Zaman & M. Oqail (2016)	Automated Service Provisioning	The key aspect associated with cloud computing is flexibility; resources can be automatically delegated or distributed [102].
	Energy Management	The benefits of energy management, which advocates cloud use, are the economies of scale [102].

TABLE 5. The summary of LB parameters.

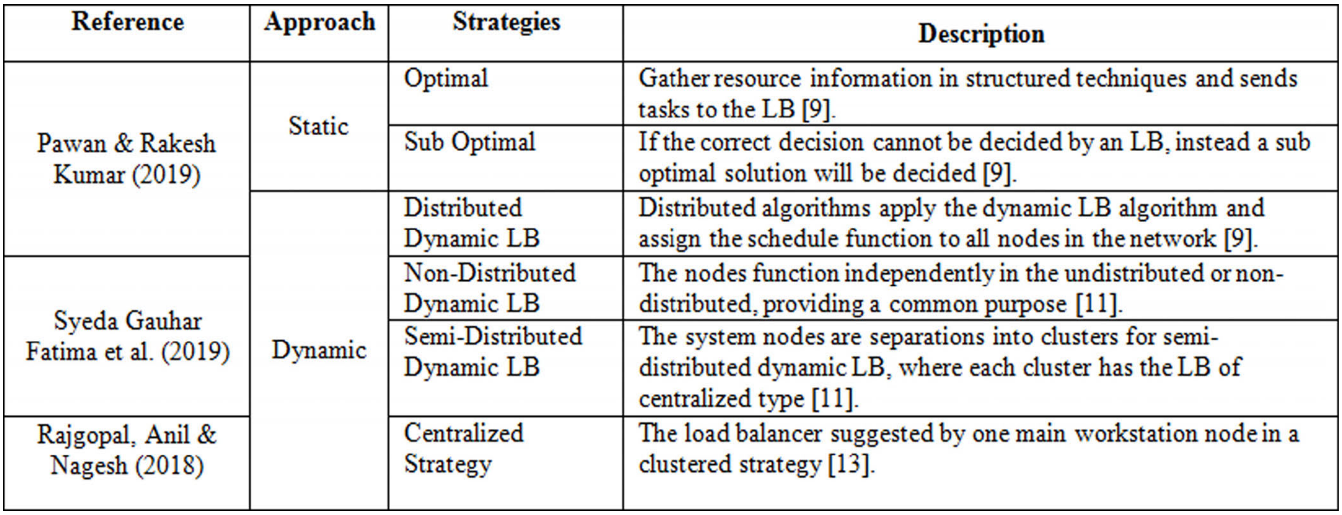


Reference	Parameters	Description
Roy Saumendu et al. (2019)	Overhead	Supports the extra cost of integrating the algorithm [64].
Shahbaz Afzal & Kavitha Ganesh (2019)	Scalability	Within the complex flow of traffic, a device can execute consumer operations [10].
	Throughput	The parameter calculates the number of activities executed in a unit of time while doing LB [10].
	Migration Time	The period of transfer is the time needed to transfer operations through imbalanced devices [10].
	Resource Utilization Factor	It reflects a portion of the services accessible for the total resources accessible [10].
	Power Saving	The metric defines the level of power & strength that the VM consumes after the process of LB is carried out [10].
	Performance	LB algorithm for the efficiency parameter increases system consistency [10].
Anurag Jain & Rajneesh Kumar (2016)	Fault Tolerance	This shows the ability of the algorithm to manage the fault situations and its strength of recovering from failures [94].
Ramandeep Kaur and Navtej Singh Ghumman (2018)	Response Time	It is counted by deducting a task's finish time from the starting time of a task's delivery [77].

finish time from the starting time of a task's delivery [77].

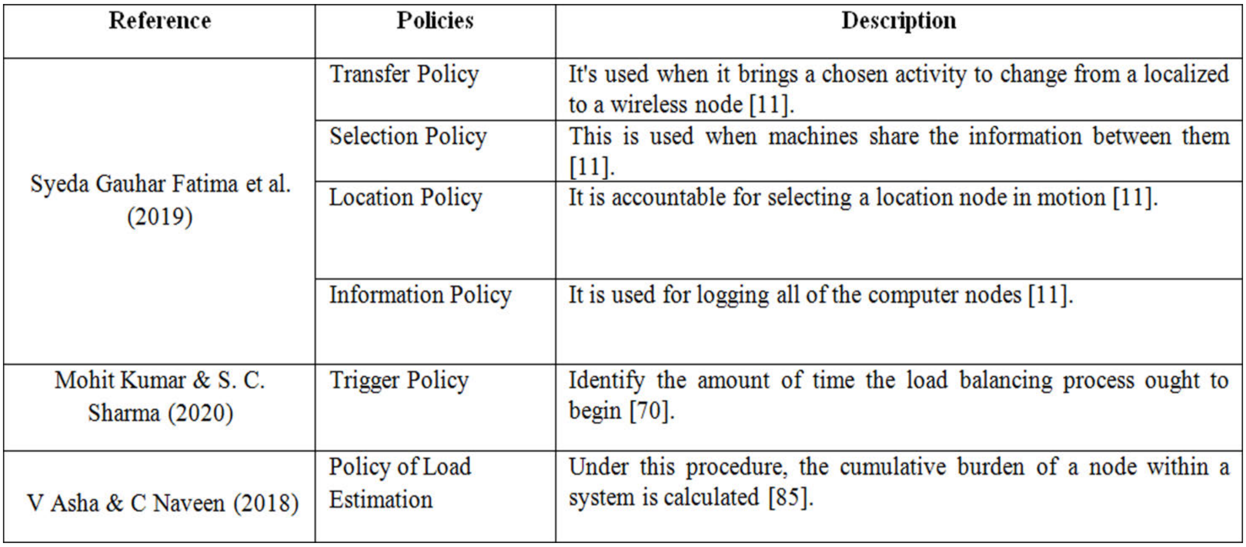
(b) **Performance:** It is the average duration that a computer needs to respond to a client order & is quantitatively equal to the sum waiting time in operation & stacked thus escaping transmission time thus retaining the efficient LB algorithm for the efficiency parameter increases system consistency. Good precision ensures a better quality of the service through SLA defense [10].

TABLE 6. The summary of LB approaches.



Reference	Approach	Strategies	Description
Pawan & Rakesh Kumar (2019)	Static	Optimal	Gather resource information in structured techniques and sends tasks to the LB [9].
		Sub Optimal	If the correct decision cannot be decided by an LB, instead a sub optimal solution will be decided [9].
	Dynamic	Distributed Dynamic LB	Distributed algorithms apply the dynamic LB algorithm and assign the schedule function to all nodes in the network [9].
Syeda Gauhar Fatima et al. (2019)		Non-Distributed Dynamic LB	The nodes function independently in the undistributed or non-distributed, providing a common purpose [11].
		Semi-Distributed Dynamic LB	The system nodes are separations into clusters for semi-distributed dynamic LB, where each cluster has the LB of centralized type [11].
Rajgopal, Anil & Nagesh (2018)		Centralized Strategy	The load balancer suggested by one main workstation node in a clustered strategy [13].

TABLE 7. The summary of LB policies.



Reference	Policies	Description
Syeda Gauhar Fatima et al. (2019)	Transfer Policy	It's used when it brings a chosen activity to change from a localized to a wireless node [11].
	Selection Policy	This is used when machines share the information between them [11].
	Location Policy	It is accountable for selecting a location node in motion [11].
	Information Policy	It is used for logging all of the computer nodes [11].
Mohit Kumar & S. C. Sharma (2020)	Trigger Policy	Identify the amount of time the load balancing process ought to begin [70].
V Asha & C Naveen (2018)	Policy of Load Estimation	Under this procedure, the cumulative burden of a node within a system is calculated [85].

C. LB APPROACHES

LB strategies are classified mainly according to the state of the operation's system & start. These are classified as the initiated sender, receiver-initiated, and symmetric based on the initiation of the procedure as described [9].

(a) **Sender Initiated:** In this method, when a node is overloaded, it looks for more nodes that are gently loaded to share a lot of work. Upon congestion of nodes, the sender initiates the process of locating the nodes under load [9].

(b) **Receiver Initiated:** In this strategy look for overloaded nodes to share a lot of work, the receiver, or lightly loaded nodes [9].

(c) **Symmetric:** Dual methods initiated by the sender and process techniques initiated by the recipient are fused into this procedure [9].

The techniques for load-balancing may be divided into the respective subcategories, depending on the system level [9]:

1) **Static:** Static LB strategies follow a static collection of rules which don't rely on the network's current state. This strategy is not scalable and involves specific knowledge of resources, such as contact time, storage and storage space of nodes, processing capacities of nodes, etc. This technique is quick and efficient, but typically the connected servers cannot be found, this leads to inconsistent resource distribution. The biggest problem with this kind of approach is that the actual state of the system is given little consideration to decision making. Therefore the constantly changing state is unacceptable for distributed systems. Static strategies only operate very well when the load variance in the nodes is lower [9]. This method's operating period is considerably less than

that of the dynamic load balancing system, or the actual work time provided by the algorithm [78]. In this algorithm, the main problem is the order constraint. Fully suited to the question for a short total time [97]. The strategies for static load-balancing are set out as [9]:

(a) **Optimal:** The DCN gathers resource information in structured techniques and sends tasks to the LB, which performs maximum allocation in a limited amount of time [9].

(b) **Sub Optimal:** If the correct decision cannot be decided by an LB, instead a sub-optimal solution will be decided. Min-Min, Max-Min, Round Robin, Shortest Job First, Two-phase Opportunistic LB (OLB), and Central LB are just some static strategies for VMs [9].

2) **Dynamic:** These approaches take the current state of the system into account, and then make decisions. The key benefit of all these strategies is that they allow the tasks to transfer from an overloaded machine to an under loaded one [9]. Dynamic load balancing resulting in tolerance to faults, higher scalability, and low overhead may be used to increase CC efficiency [81]. This was used by the dynamic load balancing approach to handling unpredictable processor loads [86]. Dynamic LB strategies are versatile which leads to improved performance. During processing, this strategy takes the steps below. It monitors the loading of the node on a regular basis. This interchange load information and state within nodes at a given time interval to calculate node workload & redistribute workload within nodes. Classify the more complex methods for load balancing as follows [9]:

(a) **Distributed Dynamic LB:** Syeda Gauhar Fatima et al [11] in the distributed algorithm apply the dynamic LB algorithm and assign the schedule function to all nodes in the network. The node relationship could adopt dual types for LB attainment: cooperative & non-cooperative.

(b) **Non-Distributed Dynamic LB:** The nodes function independently in the undistributed or non-distributed, providing a common purpose. Undistributed algorithms of dynamic LB are further broken down into two: semi-distributed and centralized [11].

(c) **Semi-Distributed Dynamic LB:** The system nodes are separations into clusters for semi-distributed dynamic LB, where each cluster has the LB of centralized type. The right election strategy selects a centralized node inside each cluster that will take care of LB within that cluster. Thus the LB of all systems occurs via the primary nodes of each cluster [11].

(d) **Centralized Strategy:** The load balancer suggested by Rajgopal KT et al [13] will be put on one main workstation node in a clustered strategy. Some main characteristics of a unified approach include:

1) The main node can contain a collection of the tasks to perform [13].
2) The activities are then forwarded to the operating node [13].
3) Upon completion of the cycle, a query for the next assignment is sent to master node [13].

D. POLICIES OF LB

Dynamic LB methods employ a policy to keep records of data changes. Many policies regulate the dynamic load balancers [11].

Each of these complex LB policies has had some partnership where the policy of change initially manages the tasks that join a system. After reading a checklist, it determines whether to transfer the tasks to a remote node or not. Location policy must specify an idle or under loaded destination node for the activities that involve a transfer. If a remote node is not yet available for execution, the job will then be put in a queue for local processing. Both the transfer policy and the location policy gather the information needed from the information policy before making the decision [9].

V. MACHINE LEARNING FOR LB

This research paper presents different ML scheduling algorithms like Simulated Annealing, Tabu Search, Particle Swarm Optimization, Fish Swarm, Optimization Algorithm, Cat Swarm Optimization Algorithm & Cuckoo Search Algorithm [36].

1) **Simulated Annealing:** Simulated annealing is motivated by annealing in solids where annealing in solids like metal or glass implies heating and allowing it to cool little by little, to erase and tighten internal stresses. Normally this method locked with local limits, undesirable allotments are trained and the algorithm also depends on the availability of requests and the bin volume. With high temperatures, the method works well [36].

2) **Tabu Search:** The tabu search algorithm is a global optimization method aimed at simulating human intellect and has a greater quality optimization capability. It is intended to direct other approaches to avoid the regional optimality trap and has been applied for solving resource allocation and other issues with optimizing [36].

3) **Partical Swarm Optimization:** Is highly sophisticated bionic heuristic, smart optimization algorithms that copy the swarm-based behavior of animals. The PSO algorithm is not efficient in solving differential restriction issues. The merits of simultaneous allocation, extensibility, simple to recognize, powerful resiliency, with excellent characteristics in dynamic environments, PSO efficiently overcomes numerous issues related to pairing optimization [36].

4) **Fish Swarm Optimization:** Is influenced by population-based meta-heuristic smart optimization algorithm combinational issues from fish swarm behaviors to solving. This method adheres to the behavior of groups of fish swarm intelligence where the community

finds a global level for the food to reach the upper concentrated areas [36].

5) **Cat Swarm Optimization:** A smart heuristic scheduling algorithm depending on cats' social behavior corresponds to the swarm intelligence family is cat swarm optimization. The result found improves the total amount of energy consumed. It also offers an optimized resource scheduling function which minimizes the scheduling costs. By decreasing the size of instances it is an enhancement over PSO [36].
6) **Cuckoo Search:** The cuckoo search technique is a meta-heuristic algorithm that models naturally occurring cuckoo genus behavior. This method provides the best remedy and efficiently balances regional and global investigation with the assistance of variable swapping. The values achieved are higher than the optimization of particle swarm [36].

VI. STUDY OF PRESENT LB METHOD

Proposed solutions for the CCLB are available in various types [11]. LB approaches are geographical distribution, general LB, natural phenomena-based LB & Network-aware task scheduling LB. Table 11 displays the synthesis of widely used LB methods for the study. Based on the study of LB's current methods, these were described as [12]:

(a) Geographical distribution based LB [11]
(b) General LB [9]
(c) Natural Phenomena-based LB [9]
(d) Network-aware task scheduling LB [9]

A. GEOGRAPHICAL DISTRIBUTION BASED LB

For the mutual production with any actual-time CC system, the proposed geographical distribution of nodes is significant, particularly for large-scale apps such as Twitter, Facebook, and so on. A very well-distributed network of cloud computing nodes assists in managing fault tolerance and enhancing application performance. Geographic LB (GLB) could be defined as a set of decisions on the digital deployment and/or relocation of VMs or computing activities to geographically dispersed data centers in order to reach SLAs or system deadlines for virtual machines/activities and to reduce operating costs for cloud systems [11].

B. GENERAL LB

In this category, the review was reviewed and overviewed in the field of general LB strategies. Although there are numerous algorithms in this group, for example, techniques such as Round Robin, Randomized Algorithm, Threshold Algorithm, OLB, OLB + LBMM, Min-Min, Max-Min, Equally Spread Current Execution Algorithm, Central LB Strategy for VMs, Throttled LB, Stochastic Hill Climbing, and Join Idle Queue [22] pros & cons are listed in Table 8.

1) **Round Robin (RR):** Kamlesh Lakhwaniin et al [14] suggested RR Algorithm Procedures are similarly selected in a particular order, Procedures are provided with a time slice during which their services are performed. Despite this, some of the assets may also be overused and some of the assets are typically idle only. HRIDYA E [15] is the static LB metric that makes the RR plot use to characterize job positions. This will carelessly pick the central hub, and then assign workplaces in a round-robin style to every other node. Nguyen Xuan Phi et al [80] is good for data centers because all Virtual Machines have the same processing power. D. Chitra Devi and V. Rhymend Uthariaraj [91] The RR strategy does not take asset capacities, importance, and task duration into account.
2) **Randomized Algorithm:** Rajgopal KT et al [13] The Selection of nodes suggested in this approach is done systematically without further knowledge of the node's present or previous load. It is suitable for situations where, because it is static, the system bears an equivalent load on each node. EYOB SAMUEL TEFERA [43] Randomized LB algorithms & with additional computational strategy, the investigator achieves minimal response time and strong resource utilization, respectively.
3) **Threshold Algorithm:** In this approach, the load will be allocated immediately once a single node is created. The selection of nodes is performed directly, without any central code being transmitted. Each node has unique replica of the load. Load characterization is categorized into under-load, medium-load & overload. Prassanna [42] The BS relates the number of HTTP requests to the threshold value when a query arrives at the cloud server. If the present rate of user queries is greater than a threshold value at the time, 'then the workload status bursts. The workload is not in good shape.
4) **Opportunistic LB (OLB):** Prakash *et al.* [16] suggested that it is a static LB algorithm so that the contemporary VM workload is no longer defined. It tries to keep every node occupied. Prajapati and Sariya [19] OLB is trying to keep every hub engaged because it doesn't know about the current workload for every system. Through a free download, OLB assigns any order to show a support center. Singh and Sohal [44] the server manages applications with a set completion schedule for each task. The system is designed to meet the needs of the client for those processes. Afterwards the cloud data centers uses the opportunistic LB algorithm to balance the load o various servers.
5) **Opportunistic LB + LB Min-Min:** Aslam and Shah [27] suggested Algorithms for Opportunist LB (OLB) & Min-min LB (LBMM) to increase task performance. This algorithm can use resources more effectively, and it increases task capability. Shah and Patel [45] OLB + LBMM Measurement follows the approach of the specialists. This comprises multiple stages in which a challenging administrator, Stage One, typically manages the workloads and

TABLE 8. General LB methods.

Method	Pros	Cons
Central Manager Algorithm [13]	It is ideal for different hosts of complex practices.	Contributes to bottlenecks, because it needs larger operation coordination.
Round Robin Algorithm [65], [4]	• Highly utilized & quick to install • Acts well with numbers of operations greater than processor numbers [4]	• Unless the servers need multiple input capabilities, overload and crash can occur • There are no hopes of having better results in the round-robin [4]
Randomized Algorithm [66]	• Making an option at random is quick. • Randomized algorithms are also better than their acceptance counterpart and quicker.	• A randomized algorithm can be very slowly unmitigated cases. • There is still a limited possibility of having the wrong responses.
Threshold Algorithm [4]	• Threshold has poor contact with inter process • A variety of allocations are made to local processes.	If all remote procedures are overrun all procedures would be manually assigned.
Opportunistic LB Algorithm [16]	Deals easily with unfulfilled obligations in sequential order to the node at present.	The tasks are planned slowly because it does not determine the node's current execution period**.**
OLB + LBMM [27]	Resources are used very efficiently and the work competency is improved.	Completion and run time of node tasks are not considered in OLB. That's why it takes a lot of time to complete the tasks.
Min-Min LB Algorithm [32]	• Is a quick and easy algorithm • Improve the overall make-span	Leads to starvation.
Max-Min LB Algorithm [32]	• Max-min greatly performs than Min-min algorithm • Small tasks are higher in number relative to long ones.	Suffers from starvation
Equally Spread Current Execution Algorithm [48]	Boost data center load times and response times	• Required to make data processing time better • Increasing the cost
Central LB Strategy for VMs [20]	Increases the performance of the whole process	Don't look at fault-tolerant systems
Throttled LB [22]	• Maximization of resource usage • Decreases average execution time dramatically	• Has not simulated in the particular situation of workload • Does not find time limits
Stochastic Hill Climbing [28]	• Tackling the issue of bottlenecks • Effective System Workload Distribution	Imperfect solution to solving problems optimization
Join Idle Queue [67]	• Low overhead connectivity • Large scalability • Low response time	• Complexity High • Homogenous resources

assigns tasks to specific nodes. It also contains a level two service manager that divides the activities into the sub-enterprises and relegates them to the operational nodes in question. It also consists of administrative nodes for performing the level three tasks.

6) **Min-Min LB:** Arul kumar and Bhalaji [17] proposed that the min-min algorithm is a basic & fast algorithm. In the first level, all the separate activities are occupied, and their estimated average production period has been determined. The processes loop until the entire workload is complete. This provided improved

productivity & response time, as well as increased asset utilization substantially, but with high overhead communication. Gopinath and Vasudevan [32] Min-Min is a quick and efficient algorithm capable of producing the best results. At first Min-Min, the optimal activities resulting in improved scheduling & overall developments of the make-span are scheduled. Their downside is first of all allocating simple tasks. Tinier tasks will then be done first, when the bigger tasks would persist in the holding stage, contributing to weak machine use. Raushan *et al.* [46] find the time of baseline fulfillment since some errands need some money. Since Min-Min initially picks the shortest errands, it stacks more of a rapidly performing asset, leaving behind inert alternatives.

7) **Max-Min LB:** Mondal and Ray [18] The suggested algorithm is almost identical to the min-min algorithm including as follows: The optimum amount is chosen after the required fulfillment duration is calculated for the work. Derakhshan and Bateni [47] CC with a scheduling algorithm for max-min is recommended. It is maintenance of the specialized algorithm Max-min. First, from all existing jobs, it gets an average run period and then picks the nearest runtime range to the average number. Often the biggest task is much too big and it's creating a machine Inconsistency. Kaur and Sharma [84], Saranya and Maheswari [99] Max-Min is closely related to the Min-Min algorithm, where jobs are chosen for the longest period.

8) **Equally Spread Current Execution:** Patel and Chauhan [31] suggested that the existing executing algorithm be shared equally, which prioritizes a job for each node. This uses the spreading spectrum methodology in which the load is distributed by perusing throughout many nodes through load volume. The load balancer transfers the job to the relevant node & achieves a higher output if the node is equipped slightly. Kaur and Mahajan [48] it is also suggested that Distribution obtain a better outcome in data centers. When adopted, the data center lead time & response time would boost efficiency. Dash *et al.* [73] in this technique, the Heap Balancer maintains a collection list of VM's and the number of requests currently allocated for the Virtual machines. Initially, all Virtual machines have 0 allots.

9) **Central LB Technique for Virtual Machines:** Ahmad and Khan [20] has suggested a Centralized LB Technique for VMs (CLBVM) that handles the cloud's job slightly this technique increases the system's overall performance but it does not check at the structures that are tolerant to weaknesses.

10) **Throttled LB:** In this algorithm, Volkova *et al.* [26] suggested that the load balancer manages a table of VM indexes, as well as their location (Accessible or Busy). The customer/host then tells the data center to find a suitable VM (VM) for the given mission. The data center requires a load balancer to handle the VM. In this algorithm, Shinde [54] the LB manages an Index table of the VMS and their (accessible or busy) location. First, the customer/host calls on the data center to find a suitable VM for the desired job. The data center needs to query for VM allocation to the load balancer. Megharaj [92] when executing the customer's order, the load balancer returned −1 to the data center if sufficient virtual machine is not identified. Panwar and Mallick [98] Hence the user first asks the load balancer to choose an optimal VM to execute the necessary activities.

11) **Stochastic Hill Climbing:** The stochastic climbing proposed by Mesbahi and Rahmani [28] is a variant of the hill-climbing algorithm which is an incomplete approach to solving issues. Since the LB algorithm depicted is distributed and thus solves the bottleneck issue, consideration has also been taken of the issue of addressing optimizing for an efficient allocation of system workload. Prasanthi *et al.* [55] Stochastic Algorithm is such a loop that moves continuously upward or backward, i.e. in the path of giant-value. If there is a greater-interest neighbor it stops.

12) **Join Idle Queue**: Kanakala *et al.* [30] this suggested algorithm facilitates massive scale shared architectures & massively distributed web services. This algorithm is an enhancement suggested for a LB simple algorithm that operates with the shared dispatchers. The optimal processor has to inform the dispatcher of their idleness in the simple algorithm without the knowledge of task applications which removes the LB function from the main path. Wang *et al.* [56] JIQ uses a variety of shared planners, each with an I-queue that holds a server idle set. Once a new object arrives on the network it faces a scheduler at random, asking to enter an idle method in their I-queue.

C. NATURAL PHENOMENA BASED LB

This section addresses several LB techniques which are influenced by natural phenomena or biological behavior, such as ant colony algorithm, genetic algorithm, honey bee foraging, artificial bee colony algorithm, hybrid (Ant colony, a honey bee with dynamic feedback), ant colony & complex network LB, osmosis LB algorithm, bee colony optimization algorithm, and LB honey bee foraging. Pros & cons are listed in Table 9.

1) **Ant Colony Optimization**: Kathalkar and Deorankar [21] suggested various ant colony algorithms also implement LB applications for the search for food. Bigger weight means the asset has higher computing power. The LB ant colony optimization not only handles the load but also reduces makes span. It is believed that all tasks are computationally intensive & separate from each other. Greco *et al.* [50] ACO is typically introduced as a powerful optimizing approach to tackle the traveling

TABLE 9. Overview of natural phenomena based LB.

Method	Pros	Cons
Ant Colony Algorithm [9], [83]	• Better LB, greater performance, reduced operation times • ACO has high effectiveness and can thus be utilized frequently in several areas [83] • ACO could be quickly coupled with the other heuristic algorithms [83]	Untested in a real cloud environment
Genetic Algorithm [22]	• Getting better efficiency of the system • Decreasing task times • Better Resource Usage	• Tiny throughput • No power saving • Missing the scalability
Honey Bee Foraging [9], [4]	• Lower makes pan and response time • Good functioning, as the complexity of the system, grows [4]	• Does not function for tasks which rely on it • As the size of the machine increases, the throughput doesn't rise [4]
Artificial Bee Colony algorithm [28]	Minimum migration time	• Don't resource utilization • Less security
Hybrid (Ant Colony, Honey Bee with Dynamic Feedback [9]	Lower response time, better using of resources, fewer task migrations	Low complexity, scalability
Ant Colony & Complex Network LB [37]	• Assign tasks to the corresponding node • Maximum job schedule	• Lack of response time • Lack of migration time • Lack of throughput
Osmosis LB Algorithm [38]	• VMs in homogenous as well as in heterogeneous settings • Helpful in reallocating the tasks between adjoining VMs	• Carry out just one task at a time • Decentralized
Bee Colony Optimization Algorithm [39]	• Minimizes scheduling times • Reducing data volume	Efficient when algorithm make span is minimal
LB Honey Bee Foraging [40]	Increase response time for VMs and decrease the make span	Does not work well without lowering the load balanced across VMs,

salesman (TSP) challenge. The key component of this form of algorithm is a coordinated agency colony named ’ants’ that continuously investigates multiple solutions into an appropriate parameter or topological storage, spreads knowledge via the disclosure of pheromone trails (such as natural food-seeking ants) and tries to determine the best way to link different sites (cities) after several decades. Greco *et al.* [50] ACO algorithm, as well as the heuristic bio-inspired optimization algorithms, have generated rising popularity in the last few years. Ragmani *et al.* [75] the ACO algorithm comprises two main stages: the pheromone trail updating & local solution building. Ye and Zhang [82] when addressing the issue of big-scale optimizations and results in weak algorithm performance, it is simple to get into localized optimal outcomes.

2) **Genetic Algorithm:** Ghomi *et al.* [22] proposed an algorithm that tries to balance the burden on cloud resources while trying to reduce the finish time for the task set in question. It is a stochastic search algorithm relying on the processes of natural selection & genetics. A simple GA consisting of triple processes: (1) availability, (2) genetic and (3) replacement operations. Farrag *et al.* [33] GA’s core is the creation of offspring via mutations & crossover methods, with the assumption that either binary coding, tree coding or numerical coding depends on the type of the chromosome. Yin *et al.* [51] Genetic algorithm is the natural calculation model of choice in simulating Darwin’s theory of biological evolution & biological method of genetic operations. Mukatia and Upadhyaya [71] the genetic algorithm reduced the period, besides raising the error rate and the magnitude of process apps. Bei and Jun [89] Genetic algorithm is a large stochastic search and optimization system motivated by Darwin’s theory of evolution.

3) **Honey Bee Foraging:** Dave *et al.* [24] suggested an Algorithm focused on bee colony optimization is proposed by imitating the behavior of honey bees, which optimizes the volume of nectar (throughput) to reach the maximum production. Gondhi and Gupta [36] Honey bees believe that they have multiple roles inside their colony over time. Active foraging bees go to a source of

food, search neighborhood resources, and gather food & back to the hive. Scout bees are studying the world around the hive, finding plentiful new food resources. At a certain given moment, a few of the foraging bees become inactive. This technique is the foraging activity that can be used in the planning of activities.

4) **Artificial Bee Colony:** Thanka *et al.* [34] suggested a Metaheuristic swarm intelligence algorithm is the Artificial Bee Colony algorithm that mimics honey bees' rummaging ways. It comprises of triple key elements: bee for staff, bee for onlookers of scouts and bee. The worker bees collect the specifics of the nectar in the dancing zone existing in the hive. Upon exchanging this knowledge again in the hive the worker bees revert to the memorized food source of the preceding process. Uma and Bala Saraswathy [74] ABC as an optimizing method offers a population-based searching technique under which entities named food locations are time-consuming changed by artificial bees and the bee's goal is to identify locations of large-nectar sources of food and ultimately the largest nectar. Tripathi *et al* [79] the area of dance is the essential proposal of hive according to the exchange of information and skills.
5) **Hybrid (Ant Colony, Honey Bee with Dynamic Feedback):** Ashouraei *et al.* [35] suggested a LB method for efficient use of CC services. The established ACHBDF process uses a hybrid technique of both dynamic scheduling methods, together with a dynamic time stage response process. Ineffective planning of the suggested ACHBDF uses ant colony process efficiency & honey bee process. For each phenomenon the response method used system load inspects in a dynamic response table to help transfer tasks more efficiently in less time. An experimental study contrasting the existing ant colony optimization, the honey bee method & showed ACHBDF's excellence.
6) **Ant Colony & Complex Network LB:** Alam and Khan [37] suggested the ACCLB algorithm to explore the shortest route between a food sources& nest. First compile all cloud server specifications in the ACCLB algorithm, to delegate activities to the correct node. If the mission is initiated from the "head processor," the onward motion initiates the ant & phenomenon in the pathway. Certify whether or not this is an overloaded processor as the ant shifts from an overloaded processor appearing for the arriving processor in the path of onward. Ultimately, if the ant discovers that it is already heading into the other path for an overloaded processor under packed processor it begins running back to the previous one under the loaded processor it had identified before.
7) **Osmosis LB Algorithm:** Mallikarjuna and Krishna [38] suggested the Osmosis LB Model (OLB) reassigns tasks to a series of VMS, with the ultimate goal of adjusting the load. The system of LB is completely decentralized & is supported by the ant-like agents that data centers conduct on a Chord overlay. Each data center interacts with a list & can execute one and more tasks simultaneously with different characteristics of the implementation.
8) **Bee Colony Optimization Algorithm:** Mallikarjuna and Krishna [39] suggested an optimization method inspired by the choice-making process for LB using an artificial bee colony algorithm. Until we go into a thorough process of LB system, let us note the bee in general. Bee colony optimization brings the deciding cycle by looking for the best food options across different opportunities. The choice-making process is dependent on the swarm.
9) **LB Honey Bee Foraging:** Mallikarjuna and Krishna [40] the suggested HBF is defined by honey bees in their search for food. There is a colony of bees foraging the supply of food. The bees create one signature dance structure called the waggle dance when there is food supply. They returned to the hive by practicing waggle dance to announce the food. It provides a sense of the duration of the dance concerning the existing amount of beehive food & distance.

D. NETWORK AWARE TASK SCHEDULING BASED LB

In this section, [22] the literature on network-aware task scheduling and LB techniques were reviewed and discussed here are some of the pros and cons set out in table 10.

1) **Shortest Job Scheduling LB Algorithm:** Gayatri Pasare et al [49] Scheduling Shortest Job First (SJF) is a priority schedule that is non-preemptive. Non-Preemptive means that, instead, the device cannot obtain the other processor until the operating cycle is complete due to the processor in real-time. Shortest function first is a complex LB algorithm that performs the job on the basis of preference. It determines preferences by deciding phase size.
2) **Task Scheduling Strategy Based on LB:** Atyaf Dhari & Khaldun I. Arif [23] proposed a work scheduling strategy based on a full allocation of resources order to enhance the Min-Min algorithm. Diminishing average execution duration to maximize resource usage. Mahfooz Alam & Zaki Ahmad Khan [37] LBVM has been started Usage of a genetic algorithm and this algorithm to improve the Min-Min algorithm based on a full resource allocation request. Succeeds great LB & decreases the migration of the dynamic VM resources. It uses recent state & historically device data needed for the smallest amount of VM assets & picks the successful solution. Despite the growth of CC, Yuanzheng Xue et al [52] continue to use a cloud service with more and more companies & individuals. The users must give SaaS providers a huge number of service requests.
3) **Active Clustering:** AR. Arunarani et al [25] proposed a heuristic, dynamic scheduling strategy for concurrent real-time jobs, implemented via a heterogeneous cluster.

TABLE 10. Description of network recognition scheduling methods for LB.

Method	Pros	Cons
Shortest Job Scheduling LB Algorithm [49]	Meet the required make span time	Can't estimate the timing of the burst
Task Scheduling Strategy Based on LB [9]	Scheduling of independent and contingent tasks, decreased response time	Fast transmission times and schedule
Active Clustering [4]	• Operates well with the capital highly used. • Use of the improved efficiency of the network to raising the throughput.	Decreases like diversity in the program rise.
Biased Random Sampling [1]	• Scalable • Reliable • LB shall be accomplished without controlling the nodes	Required multiple edges between nodes

A parallel, real-time job focused on directed acyclic graphs disembarks into a heterogeneous cluster, following a Poisson loop. When all the tasks are under their respective deadlines a work is said to be feasible. The scheduling algorithm brings stability steps into accounts and thereby increases the efficiency of heterogeneous clusters at no additional cost to the equipment. Wenzhun Huang et al [53] the independent cluster & analytical hierarchical (AHP) approach is used in tandem with hadoop to measure data to be used to have fault tolerance, reduce the time of processing & communications failures.

4) **Biased Random Sampling:** Sukrati Jain & Ashendra K. Saxena [29] the proposed comes under the dynamic LB algorithm. Therefore align the complex charge across several nodes. Analysis of the device is improved with an incredible & connected population of properties. Thus performance in a better volume is superbly optimized by using an enhanced device asset. Using this algorithm LB can be distributed effectively via computer nodes. Servers act as nodes at this level. In this process, a graph is created which represents the load of work on the nodes.

VII. FAULT TOLERANCE

FT in LB is one of the major challenges in cloud computing, which involves spreading workload uniformly to all nodes, detecting the fault and removing fault from the network, and sharing the workload to all nodes in order to maximize cloud network performance [68]. FT is a system's ability to perform its task accurately even though inner defects are present [107]. FT is one of the most key parameters because resource failure affects device efficiency, job performance, throughput, response time, and output. Thus, a policy of FT is needed to prevent failures, resolve these failures and thus improve performance metrics [60]. This research paper is the first to combine methods of FT into those three groups. There has been an increasing demand for smart systems that can learn and adjust their FT appropriately through a relationship with the environment. Upcoming guidance on cloud FT moves towards smart and resilient methods [68].

The taxonomic classification groups are FT strategies into three types, i.e. reactive methods, proactive methods & resilient methods [68].

A. REACTIVE METHODS

Reactive approaches are used to minimize the impact of errors after they have appeared [68]. The system state is stored and used consistently while restoration is undergoing operation [69].

1) **Checkpointing/Restarting:** The approach works by consistently storing system status, begin the task from the most current state in case of failure [68]. It's an approach that's effective for giant apps [95].
2) **Replication:** To render operation efficient, multiple task replicas are run on multiple resources before the entire repeated process is not crashed. Replication is implemented using HAProxy, Hadoop & AmazonEc2 [95].
3) **Retry:** The retry approach works by easily retrieving a rejected query several times over the same asset [68]
4) **Custom Exception Handling:** Includes methods where programmers inject code into the app so that during debugging they can handle different errors [68].
5) **Rescue workflow:** It allows the machine to continue working after any job fails until it can operate without remedying the fault [95].

B. PROACTIVE METHOD

The process is continually monitored, and fault forecasts are performed to mitigate the impact of faults long before they appear [68].

1) **Software Rejuvenation:** The system is scheduled for regular restarts & with a fresh state each time the program begins [95].
2) **Self-Healing:** The self-healing method is the capability of a device that enables it to identify, locate, and fix hardware and software defects efficiently [68].

TABLE 11. **Commonly used LB methods.**

Ref.	Name	Year	Techniques	General/Natural Phenomena/Network-Aware Task Scheduling
[13] [41]	Rajgopal & Nagesh Sweekriti & SudheerShetty	2018 2019	Central Manager	General
[14] [15] [80] [91]	Kamlesh, Prashant, & Munish HRIDYA E Nguyen, Cao Trung, Luu Nguyen & Tran Cong D. Chitra Devi & V. Rhymend Uthariaraj	2019 2019 2018 2016	Round Robin	General
[13] [43]	Rajgopal, Anil & Nagesh EYOB	2018 2019	Randomized	General
[13] [42]	Rajgopal, Anil & Nagesh Shenoy & Neelanarayanan	2018 2019	Threshold	General
[16] [19] [44]	A Arul, V Arul, & A Jagannathan Priyanka & Amit Nimmi & Mamta	2018 2019 2018	Opportunistic LB	General
[27] [45]	Sidra & Munam Jaimeel & Chirag	2015 2018	Opportunistic LB + LB Min-Min	General
[17] [32] [46]	V. Arulkumar Geethu Menka, Annmary, M. G. Apoorva, & N. Jayapandian	2020 2015 2018	Min-Min LB	General
[18] [47] [84] [99]	Ranjan, Payel & Debabrata Majid & Zohreh Simranjit Kaur & Tejinder Sharma D. Saranya & L. Sankara Maheswari	2016 2018 2018 2015	Max-Min LB	General
[31] [48] [73]	Niraj & Sandip Komalpreet & Rohit Nihar Ranjan Sabat	2015 2018 2019	Equally Spread Current Execution	General
[20]	M. Oqail & Rafiqul	2018	Central LB Strategy for Virtual Machines	General
[26] [54] [92] [98]	Violetta, Volkova, V. Chernenkaya & Elena Krishnanjali Geetha Megharaj & Dr. Mohan K.G. Reena Panwar & Bhawna Mallick	2018 2018 2016 2015	Throttled LB	General
[28] [55]	Mohammad reza & Amir PRASANTHI, G SRINIVASA & N Sridhar	2016 2019	Stochastic Hill Climbing	General

TABLE 11. ***(Continued.)*** **Commonly used LB methods.**

[30] [56]	V Ravi Teja, V.Krishna & K Karthik Chunpu, Chen & Julian	2015 2018	Join Idle Queue	General
[21] [50] [75] [76] [82]	Pooja & A.V. Deorankar A. Greco, A. Pluchino & F. Cannizzaro A. Greco, A. Pluchino & F. Cannizzaro Awatif, Amina, Noreddine, khalid Moussaid & Mohammed Rida Junyu Ye & Lichen Zhang	2018 2019 2019 2019 2018	Ant Colony Optimization	Natural Phenomena
[22] [33] [51] [71] [89]	Einollah, Amir & Nooruldeen Aya A, Safia & EI Sayed M Shuang , Peng & Ling Lalit Mukati & Arvind Upadhyay WANG Bei & LI Jun	2017 2015 2018 2019 2016	Genetic Algorithm	Natural Phenomena
[24] [36]	Akash , Bhargesh & Gopi Naveen & Ayushi	2016 2017	Honey Bee Foraging	Natural Phenomena
[34] [74]	M. Roshni, P. Uma & E. Bijolin Mr. R. Uma & M. BALA SARASWATHY	2019 2019	Artificial Bee Colony	Natural Phenomena
[35]	Mehran, Seyed Nima, Rachid & NimaJafari	2018	Hybrid (Ant Colony, Honey Bee with Dynamic Feedback)	Natural Phenomena
[37]	Mahfooz & Zaki	2017	Ant Colony & Complex Network LB	Natural Phenomena
[38]	B. Mallikarjuna & P. Venkata	2015	Osmosis LB	Natural Phenomena
[39]	B. Mallikarjuna & P. Venkata	2018	Bee colony Optimization	Natural Phenomena
[40]	B. Mallikarjuna & P. Venkata	2018	Honey Bee Foraging LB	Natural Phenomena
[31] [49]	Niraj & Sandip Gayatri, Anup & RashmiBhat	2015 2019	Shortest Job Scheduling LB	Network-Aware Task Scheduling
[23] [37] [52]	Atyaf & Khaldun Mahfooz & Zaki Yuanzheng, Shunfu & Xiushuang	2017 2017 2018	Task Scheduling Strategy Based on LB	Network-Aware Task Scheduling
[25] [53]	AR. Arunarani, D. Manjula & Vijayan Wenzhun, Haoxiang, Yucheng & Shanwen	2019 2017	Active Clustering	Network-Aware Task Scheduling
[29]	Sukrati & Ashendra	2016	Biased Random Sampling	Network-Aware Task Scheduling

3) **Preemptive Migration:** An app is continuously monitored and examined in this technique. Preemptive migration of a function relies on the control system for feed-back-loops [95].

4) **Prediction:** lies at the heart of the techniques for proactive FT. Faults are forecast in advance to allow the cloud system to take preventive measures to prevent or minimize the level of the failure [68].

5) **Monitoring (Feedback Loop):** Monitoring is generally used to complement other construction techniques. On an ongoing app, it is used to evaluate a set of status variables [68].

C. RESILIENT METHODS

Resilient methods allow a process to operate fulfilling client needs in the presence of malicious and to rapidly improve within a reasonable period [68].

1) **Machine Learning Approaches:** ML carries with it the smart way to FT. Through communicating via their surroundings, cloud systems are encouraged to learn, and their error solving techniques are adjusted consistently. Reinforcement Learning seems to be the most commonly used methodology in the FT environment. ML, in particular reinforcement learning, was used to incorporate or enhance a platform's FT capabilities. Such concepts can easily be added to cloud environments for apps [68].

   RL seems is the most common useful approach used in the FT discipline. ML, in particular reinforcement learning, was used to incorporate or enhance a platform's FT capabilities. Such suggestions can easily be extended into cloud domains for apps [68].

2) **Fault Induction:** Explains the concept of antifragility & application of strategies of malfunction enhancement to FT at major companies like Google and Amazon. This was accomplished using a program called GameDay [68].

   GameDay is software designed to improve durability by intentionally and at a particular time subjecting significant failures to systems to detect vulnerabilities and inter-system dependency. A GameDay practice imitates a real catastrophe, and the candidates will incorporate workers at different stages of a corporation. A GameDay experiment is only defined as effective if when the task is replicated all works perfectly. Part of the method's result is to allow organizations to learn from errors [68].

Difference between the proposed approach and the existing work using machine learning: Strategies of FT into three major categories: 1) reactive methods 2) proactive methods & 3) resilient methods. The reactive and proactive approaches are based mainly on traditional methods of FT, like replication, checkpointing, retry, monitoring, and preemptive migration [68].

The proposed solution focus to minimize the likelihood of fault existence in the system by making user job demands spread equally across existing resources [107].

For eg, several DCs that use the virtualization technology depend on Preemptive migration to handle defects triggered by server failures. There are constraints to those conventional ways. Initially, as described by their development, they are based on fixed reasoning and manage defects in a particular order. As a result, the ability to handle new faults that may arise in the future is lacking. Secondly, such implementations recognize only the inherent characteristics of the product when deciding on managing defects [68].

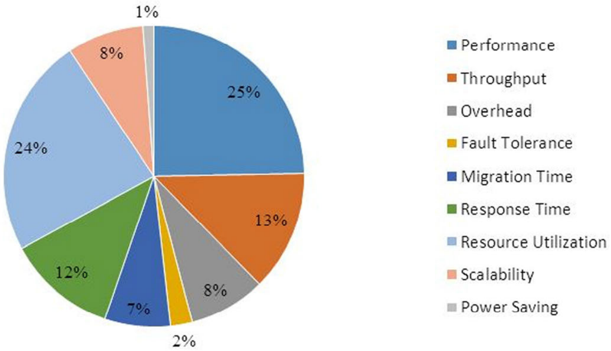


FIGURE 7. Percentage of LB parameters in the techniques analyzed.

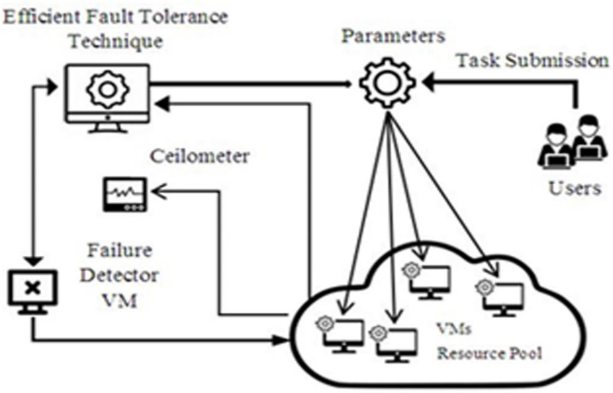


FIGURE 8. Proposed efficient FT technique for LB.

Thus, there is a need to build systems that can respond and adapt to the conditions in which they work via their connections. These systems may include the use of ML techniques as part of their remedy for FT. In this research paper, ML had been used to create solutions for FT. ML was, nevertheless, mostly used as a sub-component of the overall FT remedy. Some remedies have largely utilized ML to predict using a set of specified variables. ML has been used in many apps when handling hardware faults. In essence, such systems are adjusted and not dynamic enough to manage future defects and unidentified ones [68].

There is a need to further enhance the application of ML to FT by describing a reusable structure that can be used for managing defects in cloud environments. As a direct result of this, these agents will be permitted to make interconnect decisions which will also allow them to make ideal use of energy [68].

VIII. DISCUSSION

Table 6 provides a description of common methods for load balancing. A detailed comparison of those methods is given in Table 12 and Figure 7 shows the percentage of LB parameters in the techniques analyzed. Table 13 shows the proposed work questions focused on FT and Figure 8 proposed efficient FT technique. Table 14 shows the future of LB in cloud domain.

TABLE 12. Comparison of various LB methods, based on specific parameters.

LB Algorithm	Category	Performance	Throughput	Overhead	Fault Tolerance	Migration	Response Time	Resource Utilization	Scalability	Power Saving
Central Manager [8]	General	✓	✓	✗	✗	✗	✓	✓	✗	✗
Round Robin [8]	General	✓	✓	✓	✗	✗	✓	✓	✓	✗
Randomized [8]	General	✓	✗	✓	✗	✗	✗	✓	✗	✗
Opportunistic LB [8]	General	✓	✗	✗	✗	✗	✗	✓	✗	✗
OLB + LBMM [8]	General	✓	✗	✗	✗	✗	✗	✓	✗	✗
Min – Min [8]	General	✓	✓	✓	✗	✗	✓	✓	✗	✗
Max – Min [8]	General	✓	✓	✓	✗	✗	✓	✓	✗	✗
Central LB Strategy for VMs [8]	General	✓	✓	✗	✗	✗	✓	✓	✗	✗
Throttled [8]	General	✓	✗	✗	✓	✓	✓	✓	✓	✗
Stochastic Hill Climbing [8]	General	✓	✓	✗	✗	✗	✓	✓	✗	✗
Join Idle Queue [8]	General	✓	✗	✓	✗	✗	✓	✗	✗	✗
Ant Colony Optimization [8]	Natural Phenomena	✓	✗	✗	✗	✓	✗	✓	✗	✗
Genetic Algorithm [8]	Natural Phenomena	✓	✗	✗	✗	✗	✗	✓	✗	✗
Honey Bee Foraging [8]	Natural Phenomena	✗	✗	✗	✗	✗	✗	✓	✗	✗
Artificial Bee Colony [34]	Natural Phenomena	✓	✗	✗	✗	✓	✗	✓	✓	✓
Hybrid (Ant Colony, Honey Bee with Dynamic Feedback) [35]	Natural Phenomena	✓	✓	✗	✗	✗	✗	✗	✓	✗
Ant Colony & Complex Net Theory [8]	Natural Phenomena	✓	✓	✗	✗	✗	✗	✗	Y	✗
Osmosis LB Algorithm [38]	Natural Phenomena	✓	✓	✗	✓	✓	✗	✓	✓	✗
Bee colony Optimization Algorithm [39]	Natural Phenomena	✓	✓	✗	✗	✗	✗	✓	✗	✗
LB Honey Bee Foraging [40]	Natural Phenomena	✓	✓	✗	✗	✓	✓	✗	✓	✗
Shortest Job Scheduling [8]	Network-Aware Task Scheduling	✗	✗	✗	✗	✗	✗	✓	✗	✗
Task Scheduling Based on LB [8]	Network-Aware Task Scheduling	✓	✗	✗	✗	✗	✓	✓	✗	✗
Active Clustering [8]	Network-Aware Task Scheduling	✗	✗	✓	✗	✓	✗	✓	✗	✗
Biased Random Sampling [8]	Network-Aware Task Scheduling	✓	✗	✓	✗	✗	✗	✓	✗	✗

In CC, the toughest challenge is LB. Depending on this, the various traditional LB algorithms are not working efficiently & do not perform with Fully Performance indicators. A big problem is the Fault Tolerance (FT) performance metric in LB algorithms. The new solution needs to be changed to focus on the LB algorithm.

IX. PROPOSED WORK

FT is a major problem across CC & is one of the most critical metrics considered because resource failure affects machine & network job execution, performance, response time & quality [60].

FT LB is one of the largest challenges in CC, which includes distributing workload uniformly across all nodes, identifying faults & eliminating network faults & spreading workload to all nodes to improve cloud network efficiency [60].

Due to FT LB, a LB algorithm ought to have the FT capability, which significantly reduces the job make-span, produces efficient network & node utilization, and also achieves an III-balanced load & high system efficiency during resource loss [60].

Fault tolerance is the potential of the system to be able to proceed with its function even though a fault occurs. Tolerance in cloud faults can be accomplished by effectively managing the load received [62].

The platform's standard of fault tolerance is maintained by spreading the platform's Virtual machines on different physical hosts. The fault-tolerance stage can be represented as: if services i could usually function when ki hosts

TABLE 13. Proposed work questions.

No	Proposed work Questions	Motivation
1	What are the present FT methods based on the taxonomy of fault tolerance?	In the CC paradigm, the main problem is LB. The various traditional load balancing algorithms do not perform well and do not work based on fault tolerance performance metrics. One of its main problems in LB algorithms is the reliability parameter of fault tolerance. Concentrating on the LB algorithm requires enhancement of the effective solution.
2	Which fault tolerance method you will select to work on fault tolerance techniques?	
3	What are the existing fault tolerance techniques based on the fault-tolerance method?	
4	Which fault tolerance technique based on machine learning approaches?	
5	Which machine learning approach you will select to develop a fault-tolerance framework model?	

TABLE 14. Overcome the limitations of current algorithm.

S.No	Limitation
1.	To perform different LB operations accurately, specialized algorithms should be built to assess optimal control rates, complex thresholds, fine-grained relocation costs, contact, and data transfer times/overheads.
2.	Various core processes have their related computational overheads, like Virtual machine migration, task migration, and monitoring of device state. These are also to be carried out in a managed way.
3.	To forecast future overload/underload conditions far ahead of period with large precision, increasingly efficient workload forecasting algorithms have to be developed.
4.	LB algorithms are typically multi-objective like performance enhancement and operating cost minimization. Hence it is important to establish an effective trade-off between different conflicting goals.
5.	To check their viability for the real cloud, algorithms should be built in the real cloud world.
6.	For a detailed comparison between the existing LB methods, the performance of the LB techniques assessed concerning the standard configuration [63].

breakdown, the fault-tolerant level of protection i is specified as ki [87].

For cloud computing systems, the last way to build fault tolerance is to establish this flexibility based on a variety of policies [88].

These initiatives are split into dual proactive and reactive groups [88].

This paper proposes an efficient fault tolerance LB technique that ensures fault tolerance will properly providing multiple objectives:

(a) Performance of the system
(b) Reduces job make span
(c) Deliver effective network
(d) Node usage
(e) Fulfill III balance load
(f) Strong versatility in system
(g) Job execution
(h) Throughput,
(i) Response Time

Why Use a Machine Learning Approach to Perform Load Balancing Tasks: The use of ML methods (like supervised, unsupervised & reinforcement learning) to handle cloud.

The future path of cloud FT moves towards smart resilient methods. ML has already been implemented in a range of FT research papers to intelligence & resilience in different ways. ML brings with it the intelligent manner of doing FT [68].

RL is the most prevalent methodology in the FT field. ML, in particular RL, was used to incorporate or enhanced a service's fault-tolerance capacity. Such ideologies could be easily adapted to cloud environments for the system. By interacting with its environment, cloud systems are enabled to discover, and their fault managing techniques are adapted accordingly. There are limits to resilient techniques. Initially, as described by their implementation, they are known as fixed logic and manage faults in a particular manner. Consequently, their total absence able to manage new faults that could occur in the future. Second, these implementations only take into account the underlying system characteristics when making decisions about the managing of faults. External or internal characteristics that may impact overall performance (like temp, power, and weather) are considered very restricted. Since the future of computing is going towards the cloud, systems are exposed to failures that are not treated using conventional methods of FT. These systems will require the application of machine learning techniques as part of their solution to FT [68].

This technique combines resilient methods/adaptive methods such as machine learning and artificial intelligence have played an active role in the RSM domain. More recently, researchers have gained much more extra attention from RSMs. In cloud systems, RSMs tend to be the potential path of fault tolerance. By current definition, a system's resilience is the indicator of how easily and the system can recover quickly and continue to function properly after a system outage or failure has occurred. The loss could be due to malfunction, power failure or destruction of the devices. Typically, RSMs provide strategies that interact with the ability to respond to clients through malfunction, device status tracking, and learning ability and adjust from defects and predictions. For RSMs the system's training and adaptation are dependent on either Machine Learning (ML) or Artificial Intelligence (AI) [68].

The capacity of RSMs to assure a customer reaction is directly relevant to system consistency that forms part of Service quality (QOS). Today, many cloud faults management research activities are focused on enhancing cloud platform efficiency. Most cloud reliability work focuses on enhancements and optimization of checkpoints, space utilization and virtual machine (VM) relocation [68].

Figure 8 illustrates the whole cycle of how the proposed fault tolerance system works. The Efficient fault tolerance technique collects ceilometer data & present device node status in the pool of resources. To reduce the future anomalous behavior the effective fault tolerance technique chooses how to change the primary concern and weight valuation of each node in the pool of resources. The parameters allocate works posted due to the importance of each resource load. The failure detector unit recognizes irregularities and fault occurrences in the network during the life cycle of the system & gives signals to the defective node for retrieval mechanism. Remember that, in Figure 8, each component has its collection of functional aspects [107].

Our anomalies detector aims to avoid or identify anomalies proactively Misfits: (i) identify malicious unacceptable deterioration of the results (as a concrete anomaly) that could fail, (ii) recognize the signs and root issues of anomalous output loss to take appropriate remedial steps, utilizing Fuzzy work-sharing here, (iii) Handle the communications and connections between signs that are outward manifestations of anomalous behavior, the real issues that underlie the loss of results & Use a backup system to fine-tune future identification deficiencies and learning from authenticated outcomes to enhance effective identification of deficiencies and to keep improving implementation and implementation mechanisms through weight and primary concern modification. The following activities are taken, associated with the system of the observing, inspection, organize, and implementation control loop [107].

- **Observing:** This phase uses a ceilometer to gather information from the technique, design this information to provide a series demonstration that could be used to identify the obscured behavior in the information [107].
- **Inspection:** The major elements of that stage are to define the dependency and the relation between defects, to determine the type of failure (failure strength degree of dispersal of an anomaly within the controlled resource), and to differentiate between defect (true diagnosis of an anomaly) and disturbance (false diagnosis) [107].
- **Organize & Implementation:** This phase is related to the efficient fault tolerance technique Virtual machine for reassigning an ideal weight to the damaged elements to be capable of storing the verified routes by their correct weight [107].

The efficient fault tolerance LB technique will has various properties as follows:

(a) Systems must have been able to adapt to their environment without compromising their functionality. This is called **adaptability** [69].

(b) Any system must be in place to maintain the number of tasks performed each unit time. This is dealt with as the **Throughput**. That provides no tasks successfully performed before the fault occurs [69].

(c) The number of extra resources the machine requires to compensate from the fault. We're approaching it as being **scalable**. It has to be on the smallest list [69].

(d) To see such a system that requires lesser time to answer to customer requests is on a plus side. Discussing this portion as **response time**. This will preferably be on its lowest value [69].

(e) The number of resources that the user requires to accomplish a given task will be at its lowest rate. We treat this as **usability** [69].

(f) Another significant consideration is the no. of times a tool is open to users at the time. It is dealt with as the **availability**. This will preferably be on its higher importance [69].

(g) The use of extra resources required by the process of fault tolerance to retrieve the device from fault is addressed as **associated overheads**. Preferably it will be at its peak while the process of fault tolerance is in action [69].

X. FUTURE DIRECTIONS

Effective use of energy and computational resources has become a matter of serious concern due to the exponential growth in demands for cloud services. LB helps to boost resource efficiency, quality, and energy savings by optimal way spreading the burden in the datacenter between various computing machines. It is observed that the surveyed algorithms typically work to improve QoS, resource utilization, and energy protection. Current LB algorithms have different limitations, such as resource, energy wastage, insufficient frequency control, and static barriers. Therefore there is a lot of scope for betterment. To optimize resource efficiency, energy conservation, and output, more efficient and adaptive LB algorithms should be built to provide customers with quality services at the lowest cost [63]. Adaptive LB will allow traffic control between fast activities, efficient use of resources, and would likely involve a compounding of the centralized and distributed control mechanism. Saving energy is an important factor in providing economic growth where increased resource usage results from reduced resource collection. New methods that require load balancing based on energy consumption, carbon emissions, and support costs are therefore highly promising [67]. As a potential course, several meta-heuristics are encouraged to be tested under practical systems, such as methods rely on ACO or PSO that illustrates the possibilities to apply them in the real cloud [106]. The following work may be achieved in the future to overcome the limitations of current algorithms.

XI. CONCLUSION

This article is focused on cloud computing problems and its major challenges. Cloud computing is state-of-the-art computer technology which delivers customer support at all times. LB is one of the biggest problems with CC, as overloading a device will lead to terrible results that could create technology obsolete. So there is always a need for an effective LB algorithm for efficient use of resources. The main goal of LB is to meet user needs by distributing the workload across multiple network nodes & maximizing resource usage & growing device efficiency. Consequently, effective load management is critical for system efficiency, resource usage, reliability, throughput optimization and response time minimization. This research described the numerous algorithms for LB & their static load balancing algorithm, dynamic load balancing algorithm & dynamic nature inspired load balancing algorithm types. In the future, the need to build fully autonomous new dynamic LB algorithms will allow better use of resources, minimum make-span, and an improved degree of mismatch, effective task migrations, and minimum time span. CC itself is a technology that can last for years. It's one of the main innovations and we can use it to perform a critical part of the company. The above-mentioned innovations will make CC in the long term completely better.

REFERENCES

[1] J. Rathore, "Review of various load balancing techniques in cloud computing," *Comput. Sci. Electron. J.*, vol. 7, no. 1, p. 5, 2015.

[2] A. Garg, K. Patidar, G. K. Saxena, and M. Jain, "A literature review of various load balancing techniques in cloud computing environment," *Int. J. Enhanced Res. Manag. Comput. Appl.*, vol. 5, no. 2, p. 11–14, 2006.

[3] IJSM of and ERIjsmer. *A Review: Load Balancing Algorithm Using Cloud Analyst Environment*. Accessed: May 10, 2020. [Online]. Available: https://www.academia.edu/30460499/A_Review_Load_balancing_Algorithm_Using_Cloud_Analyst_Environment

[4] S. Joshi and U. Kumari, "Load balancing in cloud computing: Challenges & issues," in *Proc. 2nd Int. Conf. Contemp. Comput. Informat. (IC3I)*, Greater Noida, India, Dec. 2016, pp. 120–125, doi: 10.1109/IC3I.2016.7917945.

[5] A. Rashid and A. Chaturvedi, "Cloud computing characteristics and services a brief review," *Int. J. Comput. Sci. Eng.*, vol. 7, no. 2, pp. 421–426, Feb. 2019, doi: 10.26438/ijcse/v7i2.421426.

[6] U. Patel and M. H. Gupta, "A review of load balancing technique in cloud computing," *Int. J. Res. Anal. Rev.*, vol. 6, no. 2, p. 8, 2019.

[7] J. M. Shah, K. Kotecha, S. Pandya, D. B. Choksi, and N. Joshi, "Load balancing in cloud computing: Methodological survey on different types of algorithm," in *Proc. Int. Conf. Trends Electron. Informat. (ICEI)*, May 2017, pp. 100–107, doi: 10.1109/ICOEI.2017.8300865.

[8] N. Kumar and N. Mishra, "Load balancing techniques: Need, objectives and major challenges in cloud Computing- a systematic review," *Int. J. Comput. Appl.*, vol. 131, no. 18, pp. 11–19, Dec. 2015.

[9] P. Kumar and R. Kumar, "Issues and challenges of load balancing techniques in cloud computing: A survey," *ACM Comput. Surveys*, vol. 51, no. 6, pp. 1–35, Feb. 2019, doi: 10.1145/3281010.

[10] S. Afzal and G. Kavitha, "A taxonomic classification of load balancing metrics: A systematic review," in *Proc. 33rd Indian Eng. Congr.*, Jan. 2019, p. 7.

[11] S. G. Fatima, S. K. Fatima, S. A. Sattar, N. A. Khan, and S. Adil, "Cloud computing and load balancing," *Int. J. Adv. Res. Eng. Technol.*, vol. 10, no. 2, pp. 189–209, Mar. 2019, doi: 10.34218/IJARET.10.2.2019.019.

[12] B. Mallikarjuna and D. A. K. Reddy, "The role of load balancing algorithms in next generation of cloud computing," *Control Syst.*, vol. 11, p. 20, Jul. 2019.

[13] K. T. Rajgopal, K. R. A. Kumar, and N. Shenoy, "Load balancing in cloud computing: A survey on popular techniques and comparative analysis," *Global J. Comput. Sci. Technol.*, vol. 18, pp. 1–11, Jun. 2018.

[14] K. Lakhwani, "An extensive survey on load balancing techniques in cloud computinG," *J. Gujarat Res. Soc.*, vol. 21, no. 10s, pp. 309–319, 2019.

[15] E. Hridya, "Analyzing the performance of load balancing algorithms in cloud computing," *Int. J. Inf. Comput. Sci.*, vol. 6, Mar. 2019.

[16] A. A. Prakash, V. Arul, and A. Jagannathan, "A look at of efficient and more suitable load balancing algorithms in cloud computing," *Int. J. Eng. Res. Comput. Sci. Eng.*, vol. 5, no. 4, p. 7, 2018.

[17] V. Arulkumar and N. Bhalaji, "Performance analysis of nature inspired load balancing algorithm in cloud environment," *J. Ambient Intell. Humanized Comput.*, vol. 2020, pp. 1–8, Jan. 2020.

[18] R. K. Mondal and P. Ray. *Load Balancing*. Accessed: May 10, 2020. [Online]. Available: https://www.academia.edu/21990663/Load_Balancing

[19] P. Prajapati and A. K. Sariya, "A review: Methods of load balancing on cloud computing," *Int. J. Res. Anal. Rev.*, vol. 6, Mar. 2019.

[20] M. O. Ahmad and R. Z. Khan, "Load balancing tools and techniques in cloud computing: A systematic review," in *Advances in Computer and Computational Sciences*. Singapore: Springer, 2018, pp. 181–195.

[21] P. R. Kathalkar and A. V. Deorankar, "A review on different load balancing algorithm in cloud computing," *Int. Res. J. Eng. Technol.*, vol. 5, no. 2, pp. 1–3, 2018.

[22] E. J. Ghomi, A. M. Rahmani, and N. N. Qader, "Load-balancing algorithms in cloud computing: A survey," *J. Netw. Comput. Appl.*, vol. 88, pp. 50–71, Jun. 2017.

[23] A. Dhari and K. I. Arif, "An efficient load balancing scheme for cloud computing," *Indian J. Sci. Technol.*, vol. 10, no. 11, pp. 1–8, 2017.

[24] A. Dave, B. Patel, and G. Bhatt, "Load balancing in cloud computing using optimization techniques: A study," in *Proc. Int. Conf. Commun. Electron. Syst. (ICCES)*, Oct. 2016, pp. 1–6.

[25] A. Arunarani, D. Manjula, and V. Sugumaran, "Task scheduling techniques in cloud computing: A literature survey," *Future Gener. Comput. Syst.*, vol. 91, pp. 407–415, Feb. 2019.

[26] E. N. Desyatirikova, O. V. Kuripta, Y. S. Stroganova, and I. P. Abrosimov, "Quality management in IT service management based on statistical aggregation and decomposition approach," in *Proc. Int. Conf. 'Qual. Manage., Transp. Inf. Secur., Inf. Technol.' (IT QM IS)*, Saint Petersburg, Russia, Sep. 2017, pp. 500–505, doi: 10.1109/ITMQIS.2017.8085871.

[27] S. Aslam and M. A. Shah, "Load balancing algorithms in cloud computing: A survey of modern techniques," in *Proc. Nat. Softw. Eng. Conf. (NSEC)*, Dec. 2015, pp. 30–35.

[28] M. Mesbahi and A. M. Rahmani, "Load balancing in cloud computing: A state of the art survey," *Int. J. Modern Edu. Comput. Sci.*, vol. 8, no. 3, p. 64, 2016.

[29] S. Jain and A. K. Saxena, "A survey of load balancing challenges in cloud environment," in *Proc. Int. Conf. Syst. Modeling Advancement Res. Trends (SMART)*, 2016, pp. 291–293.

[30] V. R. Kanakala, V. K. Reddy, and K. Karthik, ‘‘Performance analysis of load balancing techniques in cloud computing environment,’’ in *Proc. IEEE Int. Conf. Electr., Comput. Commun. Technol. (ICECCT)*, Mar. 2015, pp. 1–6.

[31] N. Patel and S. Chauhan, ‘‘A survey on load balancing and scheduling in cloud computing,’’ *Int. J. Innov. Res. Sci. Technol.*, vol. 1, pp. 185–189, Dec. 2015.

[32] P. P. G. Gopinath and S. K. Vasudevan, ‘‘An in-depth analysis and study of load balancing techniques in the cloud computing environment,’’ *Procedia Comput. Sci.*, vol. 50, pp. 427–432, Jan. 2015.

[33] A. A. Salah Farrag, S. A. Mahmoud, and E. S. M. El-Horbaty, ‘‘Intelligent cloud algorithms for load balancing problems: A survey,’’ in *Proc. IEEE 7th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS)*, Dec. 2015, pp. 210–216.

[34] M. R. Thanka, P. Uma Maheswari, and E. B. Edwin, ‘‘An improved efficient: Artificial bee colony algorithm for security and QoS aware scheduling in cloud computing environment,’’ *Cluster Comput.*, vol. 22, no. S5, pp. 10905–10913, Sep. 2019.

[35] M. Ashouraei, S. N. Khezr, R. Benlamri, and N. J. Navimipour, ‘‘A new SLA-aware load balancing method in the cloud using an improved parallel task scheduling algorithm,’’ in *Proc. IEEE 6th Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2018, pp. 71–76.

[36] N. K. Gondhi and A. Gupta, ‘‘Survey on machine learning based scheduling in cloud computing,’’ in *Proc. Int. Conf. Intell. Syst., Metaheuristics Swarm Intell.*, 2017, pp. 57–61.

[37] M. Alam and Z. A. Khan, ‘‘Issues and challenges of load balancing algorithm in cloud computing environment,’’ *Indian J. Sci. Technol*, vol. 10, no. 25, pp. 1–12, 2017.

[38] B. Mallikarjuna and P. Venkata Krishna, ‘‘OLB: A nature inspired approach for load balancing in cloud computing,’’ *Cybern. Inf. Technol.*, vol. 15, no. 4, pp. 138–148, Nov. 2015.

[39] B. Mallikarjuna and P. V. Krishna, ‘‘A nature inspired bee colony optimization model for improving load balancing in cloud computing,’’ *Int. J. Innov. Technol. Exploring Eng.*, vol. 8, pp. 51–54, Dec. 2018.

[40] B. Mallikarjuna and P. V. Krishna, ‘‘A nature inspired approach for load balancing of tasks in cloud computing using equal time allocation,’’ *Int. J. Innov. Technol. Exploring Eng.*, vol. 8, Dec. 2018.

[41] S. M. Shetty and S. Shetty, ‘‘Analysis of load balancing in cloud data centers,’’ *J. Ambient Intell. Humanized Comput.*, pp. 1–9, Jan. 2019.

[42] J. Prassanna and N. Venkataraman, ‘‘Threshold based multi-objective memetic optimized round robin scheduling for resource efficient load balancing in cloud,’’ *Mobile Netw. Appl.*, vol. 24, no. 4, pp. 1214–1225, Aug. 2019.

[43] E. S. Tefera, ‘‘Developing wisely randomized weighted throttled load-balancing algorithm for cloud-computing environment,’’ Addis Ababa Sci. Technol., Addis Ababa, Ethiopia, Tech. Rep., 2019.

[44] N. Singh and M. Sohal, ‘‘CPU scheduling approach for cloud computing environment,’’ *Int. J. Res. Anal. Rev.*, vol. 5, no. 4, pp. i390–i393, 2018.

[45] J. Shah and D. C. Patel, ‘‘A dynamic resource mapping load balancing technique in cloud computing,’’ *Amer. J. Comput. Eng.*, vol. 2, p. 6, Dec. 2019, doi: 10.28933/AJCE.

[46] M. Raushan, A. K. Sebastian, M. G. Apoorva, and N. Jayapandian, ‘‘Advanced load balancing min-min algorithm in grid computing,’’ in *Proc. Int. Conf. Comput. Netw., Big Data IoT*, 2018, pp. 991–997.

[47] M. Derakhshan and Z. Bateni, ‘‘Optimization of tasks in cloud computing based on MAX-MIN, MIN-MIN and priority,’’ in *Proc. 4th Int. Conf. Web Res. (ICWR)*, Apr. 2018, pp. 45–50.

[48] K. Kaur and R. Mahajan, ‘‘Equally spread current execution load algorithm-a novel approach for improving data centre's performance in cloud computing,’’ *Int. J. Future Revolution Comput. Sci. Commun. Eng.*, vol. 4, no. 8, pp. 8–10, 2018.

[49] S. Seth and N. Singh, ‘‘Dynamic heterogeneous shortest job first (DHSJF): A task scheduling approach for heterogeneous cloud computing systems,’’ *Int. J. Inf. Technol.*, vol. 11, no. 4, pp. 653–657, Dec. 2019.

[50] A. Greco, A. Pluchino, and F. Cannizzaro, ‘‘An improved ant colony optimization algorithm and its applications to limit analysis of frame structures,’’ *Eng. Optim.*, vol. 51, no. 11, pp. 1867–1883, Nov. 2019, doi: 10.1080/0305215X.2018.1560437.

[51] S. Yin, P. Ke, and L. Tao, ‘‘An improved genetic algorithm for task scheduling in cloud computing,’’ in *Proc. 13th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, May 2018, pp. 526–530.

[52] Y. Xue, S. Jin, and X. Wang, ‘‘A task scheduling strategy in cloud computing with service differentiation,’’ *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 11, pp. 5269–5286, 2018.

[53] W. Huang, H. Wang, Y. Zhang, and S. Zhang, ‘‘A novel cluster computing technique based on signal clustering and analytic hierarchy model using Hadoop,’’ *Cluster Comput.*, vol. 22, no. S6, pp. 13077–13084, Nov. 2019, doi: 10.1007/s10586-017-1205-9.

[54] K. J. Shinde, ‘‘A novel approach of load balancing in cloud computing,’’ *J. Eng.*, vol. 1, p. 7, Mar. 2018.

[55] G. Prasanthi, G. S. Rao, and N. S. Babu, ‘‘A load aware matrix approach Load balancing in cloud computing,’’ *Int. J. Advance Res. Innov. Ideas Edu.*, vol. 5, no. 4, pp. 250–255, 2019.

[56] C. Wang, C. Feng, and J. Cheng, ‘‘Distributed join-the-idle-queue for low latency cloud services,’’ *IEEE/ACM Trans. Netw.*, vol. 26, no. 5, pp. 2309–2319, Oct. 2018, doi: 10.1109/TNET.2018.2869092.

[57] *Cloud Computing and Load Balancing in Cloud Computing-Survey—IEEE Conference Publication*. Accessed: May 10, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8776948

[58] Wikipedia. (Apr. 15, 2020). *Cloud Computing Security*. Accessed: May 10, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Cloud_computing_security&oldid=951138168.

[59] M. N. Birje and C. Bulla, ‘‘Cloud monitoring system: Basics, phases and challenges,’’ *Int. J. Recent Technol. Eng.*, vol. 8, no. 3, pp. 4732–4746, Sep. 2019, doi: 10.35940/ijrte.C6857.098319.

[60] S. Chamoli, D. Rana, and S. Dimri, ‘‘Fault tolerance and load balancing algorithm in cloud computing: A survey,’’ *Int. J. Advance Res. Comput. Commun. Eng.*, vol. 4, pp. 92–96, Aug. 2015, doi: 10.17148/IJARCCE.2015.4720.

[61] *Load Balancing Metrics*. Accessed: May 10, 2020. [Online]. Available: https://docs.cloud.oracle.com/en-us/iaas/Content/Balance/Reference/loadbalancermetrics.htm

[62] *Survey on Fault Tolerant—Load Balancing Algorithms in Cloud Computing—IEEE Conference Publication*. Accessed: May 10, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7124879

[63] A. Thakur and M. S. Goraya, ‘‘A taxonomic survey on load balancing in cloud,’’ *J. Netw. Comput. Appl.*, vol. 98, pp. 43–57, Nov. 2017, doi: 10.1016/j.jnca.2017.08.020.

[64] S. Roy, D. M. A. Hossain, S. Kumar Sen, N. Hossain, and M. R. Al Asif, ‘‘Measuring the performance on load balancing algorithms,’’ *Global J. Comput. Sci. Technol.*, vol. 19, pp. 41–49, May 2019. Accessed: May 10, 2020. [Online]. Available: https://computerresearch.org/index.php/computer/article/view/1833

[65] *CIT264WB Case Project 7-3: Load-Balancing Algorithms, Amanda—Security+Case Projects (5th Edition)—Site Root—Official Information Security Community for Course Technology, Cengage Learning—Featuring Mark Ciampa Blogs, Discussions, Videos, Industry Updates*. Accessed: May 10, 2020. [Online]. Available: https://groups.cengage.com/infosec2/f/20/t/2353

[66] *Google Search*. Accessed: May 10, 2020. [Online]. Available: http://cs.rkmvu.ac.in/~sghosh/public_html/nitw_igga/randomized-lecture-arb.pdf and https://www.google.com/search?sxsrf=ALeKk03pBXlc_XUOe-YoNLB2MXBYHE-YEw%3A1589133789107&source=hp&ei=3UG4XtrNBLCSlwSHgK0Q&q=http%3A%2F%2Fcs.rkmvu.ac.in%2F%7Esghosh%2Fpublic_html%2Fnitw_igga%2Frandomized-lecture-arb.pdf&oq=http%3A%2F%2Fcs.rkmvu.ac.in%2F%7Esghosh%2Fpublic_html%2Fnitw_igga%2Frandomized-lecture-arb.pdf&gs_lcp=CgZwc3ktYWIQAzIECCMQJ1DHBljHBmDcCGgAcAB4AIAB2gGIAdoBkgEDMi0xmAEAoAECoAEBqgEHZ3dzLXdpeg&sclient=psy-ab&ved=0ahUKEwiasP7X8KnpAhUwyYUKHQdACwIQ4dUDCAc&uact=5

[67] A. S. Milani and N. J. Navimipour, ‘‘Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends,’’ *J. Netw. Comput. Appl.*, vol. 71, pp. 86–98, Aug. 2016, doi: 10.1016/j.jnca.2016.06.003.

[68] M. A. Mukwevho and T. Celik, ‘‘Toward a smart cloud: A review of fault-tolerance methods in cloud systems,’’ *IEEE Trans. Services Comput.*, early access, Mar. 16, 2018, doi: 10.1109/TSC.2018.2816644.

[69] S. Kumar and D. A. S. Kushwaha, ‘‘Future of fault tolerance in cloud computing,’’ *Think India J.*, vol. 22, no. 17, p. 6, 2019.

[70] M. Kumar and S. C. Sharma, ‘‘Dynamic load balancing algorithm to minimize the makespan time and utilize the resources effectively in cloud environment,’’ *Int. J. Comput. Appl.*, vol. 42, no. 1, pp. 108–117, Jan. 2020, doi: 10.1080/1206212X.2017.1404823.

[71] L. Mukati and A. Upadhyay, ‘‘A survey on static and dynamic load balancing algorithms in cloud computing,’’ *SSRN Electron. J.*, pp. 1–8, Apr. 2019, doi: 10.2139/ssrn.3365568.

[72] A. Hota, S. Mohapatra, and S. Mohanty, "Survey of different load balancing approach-based algorithms in cloud computing: A comprehensive review," in *Computational Intelligence in Data Mining*, vol. 711, H. S. Behera, J. Nayak, B. Naik, and A. Abraham, Eds. Singapore: Springer, 2019, pp. 99–110.

[73] S. Dash, A. Panigrahi, and N. R. Sabat, "Performance analysis of load balancing algorithm in cloud computing," *Int. J. Innov. Res. Technol.*, vol. 6, no. 6, p. 11, 2019.

[74] (Jan. 5, 2019). *'Optimization Algorithms In Load Balancing: A Study,' JCA, Journal of Analysis and Computation, (An International Peer Reviewed, UGC Approved Journal)*. Accessed: May 10, 2020. [Online]. Available: http://www.ijaconline.com/optimization-algorithms-load-balancing-study/

[75] A. Ragmani, A. Elomri, N. Abghour, K. Moussaid, and M. Rida, "FACO: A hybrid fuzzy ant colony optimization algorithm for virtual machine scheduling in high-performance cloud computing," *J. Ambient Intell. Humanized Comput.*, Dec. 2019, doi: 10.1007/s12652-019-01631-5.

[76] S. K. Mishra, B. Sahoo, and P. P. Parida, "Load balancing in cloud computing: A big picture," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 2, pp. 149–158, Feb. 2020, doi: 10.1016/j.jksuci.2018.01.003.

[77] R. Kaur and N. S. Ghumman, "Task-based load balancing algorithm by efficient utilization of VMs in cloud computing," in *Big Data Analytics*, vol. 654, V. B. Aggarwal, V. Bhatnagar, and D. K. Mishra, Eds. Singapore: Springer, 2018, pp. 55–61.

[78] M. A. Mohammed, R. A. Hasan, M. A. Ahmed, N. Tapus, M. A. Shanan, M. K. Khaleel, and A. H. Ali, "A focal load balancer based algorithm for task assignment in cloud environment," in *Proc. 10th Int. Conf. Electron., Comput. Artif. Intell. (ECAI)*, Iasi, Romania, Jun. 2018, pp. 1–4, doi: 10.1109/ECAI.2018.8679043.

[79] A. Tripathi, S. Shukla, and D. Arora, "A hybrid optimization approach for load balancing in cloud computing," in *Advances in Computer and Computational Sciences*, vol. 554, S. K. Bhatia, K. K. Mishra, S. Tiwari, and V. K. Singh, Eds. Singapore: Springer, 2018, pp. 197–206.

[80] N. Xuan Phi, C. T. Tin, L. N. Ky Thu, and T. C. Hung, "Proposed load balancing algorithm to reduce response time and processing time on cloud computing," *Int. J. Comput. Netw. Commun.*, vol. 10, no. 3, pp. 87–98, May 2018, doi: 10.5121/ijcnc.2018.10307.

[81] S. Prakash, "A literature review of QoS with load balancing in cloud computing environment," in *Big Data Analytics*, vol. 654, V. B. Aggarwal, V. Bhatnagar, and D. K. Mishra, Eds. Singapore: Springer, 2018, pp. 667–675.

[82] J. Ye and L. Zhang, "The load balancing ant colony optimization based on cloud computing," in *Proc. Int. Conf. Netw., Commun., Comput. Eng. (NCCE)*, Chongqing, China, 2018, pp. 953–958, doi: 10.2991/ncce-18.2018.160.

[83] P. Xu, G. He, Z. Li, and Z. Zhang, "An efficient load balancing algorithm for virtual machine allocation based on ant colony optimization," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 12, Dec. 2018, Art. no. 155014771879379, doi: 10.1177/1550147718793799.

[84] S. Kaur and T. Sharma, "Efficient load balancing using improved central load balancing technique," in *Proc. 2nd Int. Conf. Inventive Syst. Control (ICISC)*, Coimbatore, India, Jan. 2018, pp. 1–5, doi: 10.1109/ICISC.2018.8398857.

[85] V. Asha and C. Naveen, "Load balancing in cloud computing by ant colony optimization method," *IJRTER*, vol. 4, no. 3, pp. 83–94, Mar. 2018, doi: 10.23883/IJRTER.2018.4101.SS6Y8.

[86] S.-L. Chen, Y.-Y. Chen, and S.-H. Kuo, "CLB: A novel load balancing architecture and algorithm for cloud services," *Comput. Electr. Eng.*, vol. 58, pp. 154–160, Feb. 2017, doi: 10.1016/j.compeleceng.2016.01.029.

[87] G. Gayathri and R. Latha, "Implementing a fault tolerance enabled load balancing algorithm in the cloud computing environment," *Int. J. Eng. Develop. Res.*, vol. 5, no. 1, pp. 249–249256, 2017.

[88] M. Nazari Cheraghlou, A. Khadem-Zadeh, and M. Haghparast, "A survey of fault tolerance architecture in cloud computing," *J. Netw. Comput. Appl.*, vol. 61, pp. 81–92, Feb. 2016, doi: 10.1016/j.jnca.2015.10.004.

[89] B. Wang and J. Li, "Load balancing task scheduling based on multi-population genetic algorithm in cloud computing," in *Proc. 35th Chin. Control Conf. (CCC)*, Chengdu, China, Jul. 2016, pp. 5261–5266, doi: 10.1109/ChiCC.2016.7554174.

[90] A. Kaur and P. Nagpal, "A survey on load balancing in cloud computing using artificial intelligence techniques," *Int. J. Multidisciplinary Res. Modern Educ.*, vol. 2, pp. 418–423, Dec. 2016.

[91] D. C. Devi and V. R. Uthariaraj, "Load balancing in cloud computing environment using improved weighted round robin algorithm for nonpreemptive dependent tasks," *Sci. World J.*, vol. 2016, pp. 1–14, Feb. 2016, doi: 10.1155/2016/3896065.

[92] G. Megharaj, "A survey on load balancing techniques in cloud computing," *IOSR J. Comput. Eng.*, vol. 18, no. 2, pp. 55–61, 2016.

[93] O. Kaneria and R. K. Banyal, "Analysis and improvement of load balancing in cloud computing," in *Proc. Int. Conf. ICT Bus. Ind. Government (ICTBIG)*, Indore, India, 2016, pp. 1–5, doi: 10.1109/ICTBIG.2016.7892711.

[94] A. Jain and R. Kumar, "A multi stage load balancing technique for cloud environment," in *Proc. Int. Conf. Inf. Commun. Embedded Syst. (ICICES)*, Chennai, India, Feb. 2016, pp. 1–7, doi: 10.1109/ICICES.2016.7518921.

[95] Z. Amin, H. Singh, and N. Sethi, "Review on fault tolerance techniques in cloud computing," *Int. J. Comput. Appl.*, vol. 116, no. 18, pp. 11–17, Apr. 2015, doi: 10.5120/20435-2768.

[96] F. Fatemi Moghaddam, M. Ahmadi, S. Sarvari, M. Eslami, and A. Golkar, "Cloud computing challenges and opportunities: A survey," in *Proc. 1st Int. Conf. Telematics Future Gener. Netw. (TAFGEN)*, Kuala Lumpur, Malaysia, May 2015, pp. 34–38, doi: 10.1109/TAFGEN.2015.7289571.

[97] S. Gupta and S. Sanghwan, "Load balancing in cloud computing: A review," *Int. J. Sci., Eng. Technol. Res.*, vol. 4, no. 6, pp. 1912–1916, 2015.

[98] R. Panwar and B. Mallick, "A comparative study of load balancing algorithms in cloud computing," *Int. J. Comput. Appl.*, vol. 117, no. 24, pp. 33–37, May 2015, doi: 10.5120/20890-3669.

[99] D. Saranya and L. S. Maheswari. (2015). Load Balancing Algorithms in Cloud Computing: A Review. Undefined. Accessed: May 10, 2020. [Online]. Available: https://www.semanticscholar.org/paper/Load-Balancing-Algorithms-in-Cloud-Computing%3A-A-Saranya-Maheswari/ee803ac59bb9cc17ed1fa1002df3680c9bad2743

[100] N. Taleb and E. A. Mohamed, "Cloud computing trends: A literature review," *Academic J. Interdiscipl. Stud.*, vol. 9, no. 1, p. 91, Jan. 2020, doi: 10.36941/ajis-2020-0008.

[101] *Challenges in Cloud Computing*. Accessed: May 17, 2020. [Online]. Available: https://www.tutorialride.com/cloud-computing/challenges-in-cloud-computing.htm

[102] R. Z. Khan and M. O. Ahmad, "Load balancing challenges in cloud computing: A survey," in *Proc. Int. Conf. Signal, Netw., Comput., Syst.*, vol. 396, D. K. Lobiyal, D. P. Mohapatra, A. Nagar, and M. N. Sahoo, Eds. New Delhi, India: Springer, 2016, pp. 25–32.

[103] T. C. Hung and N. Xuan Phi, "Study the effect of parameters to load balancing in cloud computing," *Int. J. Comput. Netw. Commun.*, vol. 8, no. 3, pp. 33–45, May 2016, doi: 10.5121/ijcnc.2016.8303.

[104] Jelecos. (Oct. 12, 2017). *Why it's Important to Apply Load Balancing in Your Cloud Environment*. Accessed: May 22, 2020. [Online]. Available: https://jelecos.com/uncategorized/important-apply-load-balancing-cloud-environment/

[105] *Importance of Load Balancing in Cloud Computing Environment*. Accessed: May 22, 2020. [Online]. Available: https://www.xcellhost.cloud/blog/importance-load-balancing-cloud-computing-environment

[106] M. Xu, W. Tian, and R. Buyya, "A survey on load balancing algorithms for virtual machines placement in cloud computing," *Concurrency Comput., Pract. Exper.*, vol. 29, no. 12, Jun. 2017, Art. no. e4123, doi: 10.1002/cpe.4123.

[107] H. Arabnejad, C. Pahl, G. Estrada, A. Samir, and F. Fowley, "A fuzzy load balancer for adaptive fault tolerance management in cloud platforms," in *Proc. Eur. Conf. Service-Oriented Cloud Comput.*, 2017, pp. 109–124.

[108] *Future of Cloud Computing—7 Trends & Prediction About Cloud—DataFlair*. Accessed: Jun. 22, 2020. [Online]. Available: https://data-flair.training/blogs/future-of-cloud-computing/

[109] B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions," *Future Gener. Comput. Syst.*, vol. 79, pp. 849–861, Feb. 2018, doi: 10.1016/j.future.2017.09.020.

[110] R. Buyya *et al.*, "A manifesto for future generation cloud computing: Research directions for the next decade," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–38, Jan. 2019, doi: 10.1145/3241737.

MUHAMMAD ASIM SHAHID received the Bachelor of Computer Science degree in software engineering and the Master of Computer Science degree in software engineering. He is currently pursuing the Ph.D. degree in information technology from the Malaysian Institute of Information Technology, Universiti Kuala Lumpur, Malaysia. He is also associated with the Sir Syed University of Engineering & Technology as a Faculty Member with the Computer Science Department. His research interests include machine learning, artificial intelligence, cloud computing, load balancing, and fault tolerance in reliability and availability.

NOMAN ISLAM received the Ph.D. degree in computer science from the National University of Computer and Emerging Sciences, Pakistan. He is currently an Associate Professor with Iqra University, Pakistan. He is also a Postdoctoral Researcher from the University of Kuala Lumpur, Malaysia. He has remained author of more than 60 peer-reviewed research publications in various ISI indexed journals, conferences, and books. He is the Core Faculty Member of President of Pakistan's Initiative for Artificial Intelligence and Computing (PIAIC). His research interests include key enabling technologies of 4^{th} industrial revolution, such as deep learning, cloud computing, and 5G networks.

MUHAMMAD MANSOOR ALAM received the M.S. degree in system engineering and the M.Sc. degree in computer science from France, U.K., and Malaysia, and the Ph.D. degree in computer engineering and the Ph.D. degree in electrical and electronics engineering. He is currently a Professor of computer science. He is also working as an Associate Dean with CCSIS and the HOD of the Department of Mathematics, Statistics, and Computer Science. He is enjoying 20 years of research and teaching experience in Canada, England, France, Malaysia, Saudi Arabia, and Bahrain. He has authored more than 150 research articles which are published in well reputed journals of high impact factor, Springer Link book chapters, Scopus indexed journals, and IEEE conferences. He has honor to work as an online laureate (facilitator) for MSIS program run by Colorado State University, USA, and Saudi Electronic University, Saudi Arabia. He has also established research collaboration with Universiti Kuala Lumpur (UniKL) and Universiti Malaysia Pahang (UMP). He is also working as an Adjunct Professor with UniKL and supervising 12 Ph.D. students. He has done Postdoc from Malaysia in Machine Learning Approaches for Efficient Prediction and Decision Making. Universite de LaRochelle awarded him Très Honorable (Hons.) Ph.D. due to his research impact during his Ph.D.

MAZLIHAM MOHD SU'UD received the Diploma degree in science, the bachelor's degree in electronics electrotechnics and automation, the master's degree in electronics electrotechnics and automation, and the Post Master degree in electronics from the University de Montpellier II, France, the master's degree in electrical and electronics engineering from the University of Montpellier, in 1993, the Ph.D. degree in computational intelligence & decision from the University De La Rochelle, France, and the Ph.D. degree in computer engineering from the Université de La Rochelle, in 2007. Since 2013, he has been the President/CEO of Universiti Kuala Lumpur, Malaysia. He has vast experience of publishing in high quality international scientific journals and conference proceedings. He has numerous years' experience in industrial and academic field.

SHAHRULNIZA MUSA received the bachelor's degree in science—physics, informatics and electronics from the Université de Metz, France, in 1992, the master's degree in education (technical and vocational education and training) from Universiti Technology Malaysia, in 2000, specialising in TVET curriculum development, and the Post-Graduate Diploma degree in integrated research study and the Doctor of Philosophy (Ph.D.) degree in communication network security from the Faculty of Electrical and Electronic Engineering, Loughborough University, U.K., in 2005 and 2008, respectively. He started his career at Universiti Kuala Lumpur (UniKL), since its establishment in 2002. From 2014 to 2017, he has served as the Dean of the Malaysian Institute of Information Technology, UniKL. He is currently a Full Professor and the Deputy President in charge of Academic and Technology of UniKL. His research interests include cybersecurity, the IoT application, the IoT security, bigdata analytic, and SDN. Apart from teaching and post-graduate supervision, he is also active in software project consultation and development in Business Application, Enterprise Resource planning (ERP), and Customer Relation Management (CRM).