Comprehensive review of load balancing in cloud computing system

Mayowa O. Oyediran[1], Olufemi S. Ojo[1], Sunday Adeola Ajagbe[2,3], Olukayode Aiyeniko[4], Princewill Chima Obuzor[5], Matthew Olusegun Adigun[2]
[1]Department of Computer Engineering, Ajayi Crowther University, Oyo, Nigeria
[2]Department of Computer Science, University of Zululand, Kwadlangezwa, South Africa
[3]Department of Computer Engineering, First Technical University, Ibadan, Nigeria
[4]Department of Computer Science, Lagos State University, Ojo, Nigeria
[5]School of Science, Engineering and Environment, University of Salford, Salford, United Kingdom

Article Info

Article history:

Received Jan 2, 2024
Revised Feb 5, 2024
Accepted Feb 9, 2024

Keywords:

Algorithm
Cloud
Cloud computing
CloudSim
Load balancing

ABSTRACT

Load balancing plays a critical role in optimizing resource utilization and enhancing performance in cloud computing systems. As cloud environments grow in scale and complexity, efficient load balancing mechanisms become increasingly vital. This paper presents a comprehensive review of load balancing techniques in cloud computing systems, with a focus on their applicability, advantages, and limitations. The review encompasses both static and dynamic load balancing approaches, evaluating their effectiveness in addressing the challenges posed by cloud infrastructure, such as heterogeneity, scalability, and variability in workload demands. Furthermore, the review examines load balancing algorithms considering factors such as resource utilization, response time, fault tolerance, and energy efficiency. Additionally, the impact of load balancing on cloud performance metrics, including throughput, latency, and scalability, is analyzed. This review aims to provide insights into the state-of-the-art load balancing strategies and serve as a valuable resource for researchers, practitioners, and system designers involved in the development and optimization of cloud computing systems.

Corresponding Author:

Sunday Adeola Ajagbe
Department of Computer Engineering, First Technical University Ibadan
Km 15 Lagos–Ibadan Expy, 200261, Ibadan, Oyo, Nigeria
Email: saajagbe@pgschool.lautech.edu.ng

1. INTRODUCTION

Every day of our lives, humans require access to four basic utility services: electricity, water, fuel, and telephone. These services must be available around the clock so that users may use them whenever convenient. John McCarthy had the vision in the 1960s that computers would become the fifth utility after the ones described above and would offer the people the fundamental computing services, they would need to satisfy their daily requirements [1]. In the twenty-first century, the computer industry will undergo a change that will make it possible for the general population to use computer services on demand. Following the creation of the internet, the internet of things (IoT) came into being as the first technology to connect people with the virtual world.

However, inspiration from the IoT model recently gave birth to the fifth novelty in the information technology (IT) industry, named cloud computing. It enables an impactful IT transformation through the provision of dynamic resource allocation to users [2]. Cloud computing has progressed from earlier iterations of grid and middleware computing technologies. It involves clusters of computers in geographically distributed networks that offer dynamic provisioning capabilities, which can be tailored to meet the specific requirements of users. In spite of the common characteristics of cloud and grid computing, cloud computing offers distinct advantages that surpass those of grid computing which is a form of distributed computing that involves pooling together resources from multiple computers to work on a task. The ability of virtualization in cloud computing can be seen as a culmination and advancement over grid computing. Cloud computing is commonly perceived as the integration of scalable and flexible virtualized resources, capable of dynamic provisioning over the internet.

This study comprehensively reviewed load balance in the cloud computing system, with a view to examining techniques available for load balancing in cloud. Below is a summary of this review paper's main contributions: i) provides comprehensive knowledge on cloud computing, ii) existing specifics on the load balancing strategy, iii) summary of major contributions to the cloud computing load balancing approach, and iv) current study focus on cloud computing's load balancing strategy.

The paper has been organized as follows: overview of cloud computing systems, potentials of cloud computing in section 2, load balancing in cloud computing in section 3, overview of load balancing techniques in cloud computing, algorithms for load balancing in cloud computing, evaluation metrics of load balancing, through the state-of-the-art review in section 4 and conclusion in in section 5.

2. CLOUD COMPUTING SYSTEM

This section reviewed the overview of cloud computing and its potential in cloud computing, highlighting the benefits such as It helps your audience understand why they should care about what you're presenting, and it emphasizes the positive outcomes or advantages. You can influence their decision-making process in your favor. Challenges and establishing research areas in cloud computing are also discussed in this section. We explained cloud computing layers, cloud computing taxonomy, and possible applicability of cloud computing were also provided according to the existing literature.

2.1. Overview of cloud computing

The term "cloud computing" is a combination of two distinct words, namely "cloud" and "computing". The "cloud" represents an infrastructure comprising both software and hardware, offering services to end users. The efficient utilization of these infrastructures is achieved through effective resource management [3]. Cloud computing can be likened to "utility computing" that provides various services across infrastructure, platform, and software. According to the National Institute of Standards and Technology (NIST), cloud computing is defined as a shared resource pool that can be easily configured, provisioned, and released with minimal effort from clients and service providers [4], [5]. Virtualization, availability and scalability of cloud computing are the distinct attributes that differentiate it from cluster and grid computing. Whereby, availability, accessibility, resource sharing, elasticity and Pay-As-You-Go are the major characteristic of cloud computing [6], [7].

a. Availability: users may simply allocate and remove resources without experiencing excessive waits.
b. Accessibility: users can use laptops, smartphones, and desktop computers to access cloud services.
c. Resource sharing: the cloud is a collection of shared resources that a number of users share.
d. Elasticity: users can scale up/down their networks based on their requirements.
e. Pay-As-You-Go: users are charged based on their resource use.

Cloud computing resources can be accessed through various methods, known as the cloud computing layer, as illustrated in Figure 1. At the uppermost level of this layer lies "software-as-a-service" (SaaS). This model focuses on providing applications to end-users through a web-based platform. In the field of biomedicine and healthcare, there is a growing trend towards adopting this model for numerous applications [8]. The purpose of this layer is to shield users from complexities. From the user's viewpoint, there is no need to be concerned about servers, networks, or the intricacies of application deployment. Distributing software as a service offers numerous advantages, with one significant benefit being the ability to provide enterprise applications without the need for software deployment on local infrastructure. Platform-as-a-service (PaaS) represents a lower-level component of the architecture, enabling developers to design and deploy new applications by leveraging the provider's application programming interface (API) [9].

PaaS serves as a platform that encompasses a set of developer tools for creating applications, developing web graphical interfaces, and designing databases. These applications and services are then delivered to a remote hosted platform offered by a cloud provider, which is supported by robust hardware infrastructure. PaaS provides developers with all the necessary resources and capabilities to create and deploy web applications and web services over the internet, utilizing cloud services [10]. The third layer is known as "infrastructure-as-a-service" (IaaS), which offers resources like virtualized servers, network devices, and various other resources. For example, Amazon EC2 is categorized as IaaS. The IaaS layer holds significant importance for cloud providers since both the PaaS and SaaS layers rely on it. Therefore, evaluating the performance of IaaS is crucial as it directly influences the performance of the other two layers. These three layers work together to provide a comprehensive experience in cloud computing [11]. Cloud computing can be categorized into four groups namely as presented in Figure 2: public cloud, private cloud, hybrid cloud and community cloud.

a. Public cloud: the public cloud empowers end-users to access a diverse range of services and resources via the internet. While many services in the public cloud are available free of charge, there may be instances where users are billed based on a "pay-as-you-go" model.
b. A private cloud refers to cloud infrastructure dedicated solely to a single consumer, such as an individual organization that may include multiple business units. This cloud infrastructure is owned, managed, and operated either internally by the organization, by a third-party, or through a combination of both. It can be hosted either internally within the organization's premises or externally [12].
c. Hybrid cloud is characterized as a blend of an organization's privately owned infrastructures along with the flexibility to leverage the advantages of the public cloud when required.
d. The community cloud model involves multiple organizations within a specific community sharing a common cloud infrastructure [13]. These organizations typically share common concerns such as mission objectives, security requirements, compliance, and jurisdiction. While the costs of this cloud model are distributed among a smaller number of users compared to the public cloud, they are still shared among a larger user base compared to the private cloud [14].

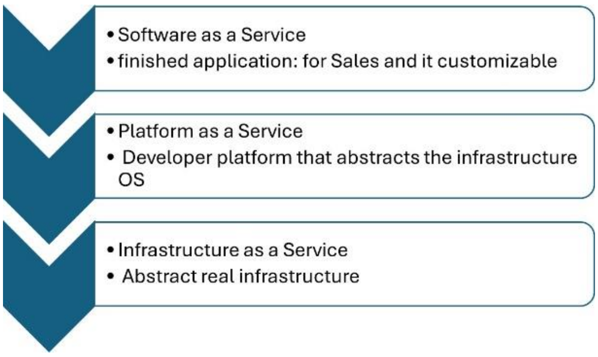


Figure 1. Cloud computing layers

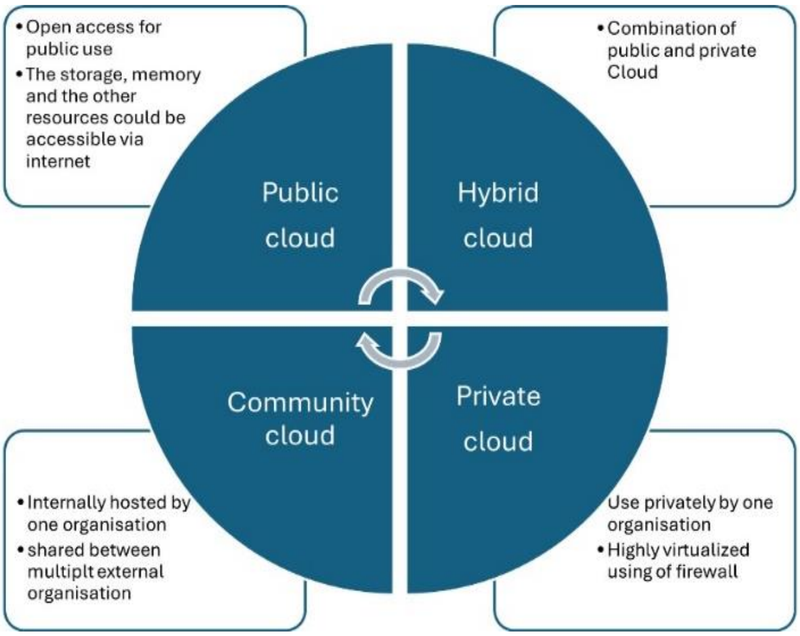


Figure 2. Cloud taxonomy [15]

2.2. Potentials of cloud computing

Cloud computing provides a wide range of advantages for users. The unlimited availability of resources, advantageous payment models, and the scalability and elasticity features of cloud networks have encouraged businesses to transition from their traditional legacy systems to innovative cloud infrastructure. Some of the benefits are presented in Table 1. Also, as promising as cloud computing sounds to be, it suffers from some challenges which are presented in Table 2.

Table 1. Benefits of the cloud computing [16]

Benefit	Opportunity
Scalability	Add or remove resource or demands
Elasticity	Allocate and release resources upon usage
Mobility	Accessing the cloud network is not dependent on time and location
Low infrastructure cost	Helps small businesses to grow sooner
Latest updates in IT	Regular updates of the system, implemented by cloud providers
Increased data storage	Access to large storage capacity for data storage and backup plan
Disaster recovery	Using the virtual backups, recovery will be 4 times faster than using another system than cloud
Availability	Rapid deployment of the infrastructure to access the resources
Billing and payment	Users will be charged per services based on their usage.

Table 2. Applicability of cloud computing [17]

Research area	Description
Interoperability	Lack of standards for service portability between cloud providers
Security and privacy	Lack of improved techniques in authorization and authentication for accessing the user’s information
Resiliency	The ability of the system to provide users with standard level of services while experiencing faults and challenges in the system
Reliability	The chance of failure in standard period of time
Energy saving	Defining a standard metric for effective power usage and an efficient standard of infrastructure usage
Resource monitoring	Lack of accurate monitoring mechanism using sensors to collect the data from CPU load and memory load.
Disaster recovery	Using the virtual backups, recovery will be 4 times faster than using another system than cloud
Load balancing	Lack of standard way of load monitoring and load management for different cloud applications

3. LOAD BALANCING IN CLOUD COMPUTING

This section discussed one of the most common research areas in cloud computing, with interest in the overview of load balancing, load balancing techniques and algorithms for load balancing in cloud computing. Load balancing and elastic scalability play a crucial role in both cloud computing and other computing systems. Load balancing is commonly employed to manage incoming traffic and prevent system failures by redirecting workloads away from overloaded and unresponsive resources. This characteristic draw influence from grid computing. Load balancing can be defined as the process of redistributing the overall workload across all nodes or systems within a distributed system. The objective is to optimize response time, enhance resource utilization, and prevent any individual node from becoming overloaded. There is nothing like the generally acceptable definition of load balancing because researchers view it from different perspectives and backgrounds. However, no matter the definition giving to load balancing, the following goals must be listed and achieved: i) reducing job response time while keeping acceptable delays, ii) maintaining system stability, iii) having fault tolerance ability, iv) improving the general system performance for achieving optimal resource utilization, maximum throughput and avoiding overload, and v) improve and maintain the availability in cloud system.

Ajagbe *et al.* [14] describe load balancing as the process used to regulate incoming traffic and stop providing tasks to overburdened and unresponsive resources. It was also proposed that load balancing methods must be resilient enough to be compatible with a wide range of application types. The following checklist is recommended for designing more effective algorithms for load balancing:

a) Complexity: the algorithm must be intelligent and simple. Complexity may increase an algorithm's overhead. Consequently, more
b) Scalability: the algorithm should be adaptable to the network expansion and abridgement.
c) Fault tolerance: the algorithm must be intelligent and simple. Complexity may increase an algorithm's overhead. Consequently, more
d) Performance and makespan: the load balancing algorithm should optimize the cloud performance by minimizing the total execution time.

3.1. Load balancing techniques in cloud computing

Load balancing technique is categorized into two main groups namely: static and dynamic technique. The two main groups of load balancing techniques are discussed as follows:

a) Static load balancing: In a static environment, it is necessary to have prior knowledge about the capacities, processing power, memory, and performance of each node. The statistical requirements cannot be modified during runtime. While implementing load balancing algorithms in a static environment is relatively straightforward, it may not be suitable for heterogeneous computing models. An example of a simple static algorithm is the Round-Robin algorithm. In this resource scheduling method, tasks are served in the order they arrive, and the least-loaded resource is allocated to complete the tasks based on time sharing principles [18].

b) Dynamic load balancing: In this environment despite the need for prior, like static environment, the algorithms operate according to the run-time statistics. These load balancing algorithms are more flexible to change and they are highly adaptable to cloud environments. Each of the static or dynamic algorithms could be divided into four (4) different categories, Figure 3 depict all the algorithms.
   - Centralized vs. distributed: the centralized model is designed with a central controller and the capacity to retain all resource information centrally. However, the model is not scalable in terms of the network. Distributed load balancer, on the other hand, lacks a centralized controller, but performs well in cases of scalability and fault tolerance, thus it may enable elasticity.
   - Preemptive algorithm vs non-preemptive: preemptive algorithms permit interrupts during execution. For instance, based on the task priority, the algorithm might cease reordering the jobs in its queue. In contrast, non-preemptive algorithms prohibit disruptions until all allocated tasks have been planned on the available resources [17].
   - Immediate vs batch mode: in immediate mode, as soon as the tasks are assigned for scheduling, they will be sent to the queue. In batch mode, tasks should be grouped based on the criteria then the group of tasks will be sent to the scheduling queue.
   - Independent vs workflow: before assigning tasks to available resources, workflow modelling should explore interdependencies between tasks. The popular languages that may depict workflow scheduling methods are directed acyclic graph (DAG) and Petri nets. However, independent modelling will arrange the jobs without considering their interdependence [19].

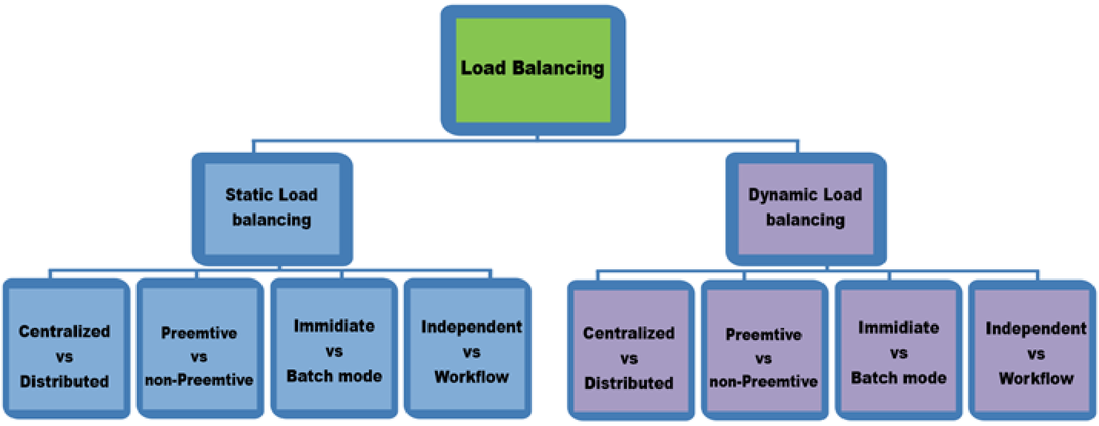


Figure 3. Load balancing technique categories [18]

3.2. Algorithms for load balancing in cloud computing

Independent and dependent scheduling methods are considered as the key approach in terms of load management. Dependent scheduling, however, is attracting more attentions [6]. Dependent scheduling is suitable for those types of tasks with dependent structured patterns. In these models, each job is composed of several dependent tasks. Therefore, execution of one task is dependent on another [20]. Unlike the independent tasks, failures in dependent-tasks execution affect the performance of the whole system. Currently, there is a lack of effective algorithm that deals with load balancing on dependent structured jobs [4], [5].

According to the classification of dependent scheduling algorithms as NP-hard issues, there is no universal algorithm that can produce an ideal solution for all scheduling problems. The non-deterministic polynomial is abbreviated as NP. A class of problems known as "NP-complete" lacks an effective technique for locating an ideal solution. In other words, no algorithm exists that can handle the problem's growing number of steps at a polynomial pace. The subset of NP issues that cannot be solved by non-deterministic machines is known as the NP-hard problem [4]. Dependent algorithms are primarily made to accommodate the Cloud infrastructure's vast distributed collaborative e-business and e-science application workloads even in discrete particle swarm optimization (DPSO). The following categories apply to dependent scheduling algorithms: i) best-efforts (heuristic) scheduling algorithms and ii) the quality of service (QoS) constraint scheduling algorithms.

The goal of the best effort algorithms is to reduce the execution time. However, they do not work with the pay-as-you-go charging model [6]. On the other hand, QOS-based approaches are regarded to be advantageous for high-quality applications. These kinds of tasks will be structured in accordance with the requirements for quality [20]. Tables 3 and 4 list the dependent and independent scheduling techniques along with a summary of the advantages and challenges of each of the stated scheduling algorithms.

Table 3. Summary of the reviewed dependent scheduling algorithms [21]

Load balancing algorithm	Static/Dynamic	Benefits	Challenges
HEFT	Static	Makespan will be improved	Scalability and resource utilizations are not considered within this algorithm
SHEFT	Dynamic	Optimize execution time	Overhead will be added to pre-calculating the start/finish time for the tasks.
Transaction-intensive cost constrained	Static	Minimize cost based on considered time.	Only will consider the in-time task scheduling processes.
QoS heuristic workflow scheduling	Static	CPU utilization is achieved through applying different parameters of QoS.	Trustworthiness and fault tolerance are not considered in this algorithm.
A practical swarm optimization-based heuristic	Dynamic	Makespan will be minimized.	Scalability and fault tolerance is not considered.
Deadline constrained workflow scheduling in software as a service.	Dynamic	Minimize cost while meeting the deadlines. Minimize makespan.	Complex algorithm. Computation cost of the algorithm is high.
Deadline and budget constraint	Static	Suitable for large scale distributed systems.	Do not consider communication and execution of the tasks.
Dynamic critical path	Dynamic	Tasks will be prioritized base on their completion time.	Do not consider scalability and fault tolerance.
Discrete particle swarm optimization	Dynamic	Minimize cost. Better performance on makespan.	Fault tolerance is not considered in this algorithm.
Genetic algorithm (GA)		Minimizing the makespan and costs	Fault tolerance is not considered.

Table 4. Summary of the reviewed independent scheduling algorithm [21]

Load balancing algorithm	Static/Dynamic	Benefits	Challenges
Round Robin	Static	Distribute the traffic evenly based on time slices.	Real-time load cannot be considered. Longer waiting time.
Dynamic Round Robin	Dynamic	Minimize the waiting time. Minimize response time.	The performance of the algorithm is low.
Signature patterning	Dynamic	Resource status and allocation are managed precisely.	Extra overhead will be added due to pattern capturing and comparison
Task Consolidation algorithm	Dynamic	Cost effective. Energy efficient.	It could be applied on local clouds.
Replication algorithm	Dynamic	Improve access time.	Does not support pre-replication and fault tolerance
Map reduce	Dynamic	Suitable for large distributed networks	Due to parallel processing, nodes can be overloaded.
Ant colony	Dynamic	It uses meta-heuristic approach.	In-effective resource utilization is needed.
Min-Min		Improve availability.	Cannot support fault tolerance
Max-Min		Minimize the waiting time.	Cannot support fault tolerance
Artificial bee colony	Dynamic	Suitable for scalability	System throughput cannot be fully optimized.
DPSO		Improved availability of the resources. Suitable for elasticity and scalability purposes.	Fault tolerance can be supported. Resources cannot be fully optimized.

3.3. Evaluation metrics of load balancing

Load balancing is essential in cloud computing to uniformly distribute the active workload across all nodes. By ensuring a well-planned and equitable distribution of resources inside the cloud computing environment, it aids in obtaining a high user satisfaction and resource utilization ratio. A good load balancing strategy helps with resource use reduction, fail-over implementation, scalability, preventing bottlenecks and over-provisioning. This section looked at a variety of assessment measures for load balancing in cloud computing, including:

Throughput is used to calculate the number of information or tasks whose execution has been completed. It is a measurement of the number of information units that a system can process in a specified length of time. It covers a wide range of systems, from organizations to different parts of computer and network systems.

- Overhead: to properly distribute the active burden across all of the nodes in cloud computing, load balancing is essential [22]. By confirming an orderly and equitable distribution of resources inside the cloud computing environment, it aids in obtaining a high user fulfilment and resource utilization ratio. A good load balancing strategy may help with a variety of tasks, including lowering resource use, establishing fail-over, enabling scalability, preventing bottlenecks and over-provisioning. The following list of assessment metrics for load balancing approaches in cloud computing was taken into consideration in this section.

Fault tolerance is the system's or algorithm's capacity to carry out accurate load balancing in the case of connection loss. A good fault-tolerant approach should be the load balancing algorithm or technique. It is a procedure that gives an operating system the ability to react to a software or hardware malfunction. According to this definition of fault-tolerance, the system's capacity to function even in the face of malfunctions or failures.

- Migration time: the execution time required to load all necessary code units on the destination host, the agent data and state, and the transmission delay between the two hosts all affect how long it takes to go from the home host to the first host on the itinerary.
- Response time: this is the measure of total elapsed time between the start and completing of a task or request. Response time measures the time taken for scheduling algorithm to respond to a job. Resource Utilization is used to check the exploitation of resources. Scalability is the ability of an algorithm to measure according to the requirement. Performance is used to check the efficiency of the system. This has to be improved at a reasonable cost: reduce task response time while keeping acceptable delays.

4. STATE-OF-THE-ART REVIEW

A number of related research have been carried out on cloud computing using different approaches; some of the areas include security, migration process and load balancing, discussed below are some of the related works in terms of load balancing. Adekunle *et al.* [23] presented a Dynamic Round Robin for load balancing in a cloud computing, the researcher studied the effect of Round robin technique with dynamic approach by varying the vital parameters of host bandwidth, cloudlet long length, virtual machine (VM) image size and VM bandwidth. Load was optimized by setting dynamic round robin by proportionately varying all these parameters. Simulator CloudSim was used for the implementation, results obtained show that system dynamically distribute load across the itinerary. Kanbar and Faraj [24] introduced a better load balance model for the public cloud based on the cloud partitioning concept with a switch mechanism to choose different strategies for different situations. The algorithm applied game theory to the load balancing strategy to improve efficiency in the public cloud environment. The model developed was compared with other ones and it was discovered that the system was able to partition a public cloud and also can switch between two different strategies, but the system suffers from optimized resource utilization.

Nadaph and Maral [25] developed partitioning algorithm and load balancing algorithm for cloud computing the paper reveals the method by which a cloud can be partitioned and a study of different algorithm with comparative study to balance the dynamic load. The comparative study between ant colony and honey bee algorithm gives the result which algorithm is optimal in normal load condition also the simplest round robin algorithm was applied when the partitions are in idle state. In the results obtained, ant colony outperforms honey bee when compared. However, the algorithm suffers from non-dynamic partitioning of resources. Husain *et al.* [9] gives an efficient dynamic load balancing algorithm for cloud workload management by which the load can be distributed not only in a balanced approach, but also it allocates the load systematically and uniformly by checking certain parameters. It balances the load on the overloaded node to under loaded node so that response time from the server was decreased and performance of the system was increased. The authors focused on response time without any interest in the waiting time of jobs.

Heidari *et al.* [1] proposes a multi objective hybrid ant colony optimization-particle swarm optimization (ACO-PSO) optimization algorithm for minimizing resource wastage, minimizing power consumption and for load balancing in physical servers by combining ACO and PSO meta heuristic algorithm. Simulation results showed that the proposed algorithm reduced resource wastage and power consumption and also provides load balancing in servers when compared to the existing multi-objective ant colony system algorithm. Xiao *et al.* [26] proposed an autonomous agent-based load balancing algorithm (A2LB) which provides dynamic load balancing for cloud environment.

Liang and Yang [27] developed a GA based load balancing strategy for cloud computing, the strategy is a neoteric load balancing strategy using GA. The algorithm thrives to balance the load of the cloud infrastructure while minimizing the make span of a given tasks set. The developed load balancing strategy was simulated using the cloud analyst simulator. The simulation results for a typical sample application shows that the developed algorithm outperformed the existing approaches like first come first serve (FCFS), Round Robing (RR) and a local search algorithm. The strategy makes use of network parameters of task to decide where to forward traffic which result in application unresponsiveness.

Abdullahi and Ngadi [7] developed a hybrid symbiotic organisms search (SOS) optimization algorithm for scheduling of tasks on cloud computing environment to optimize task scheduling in cloud computing environment based on a proposed simulated annealing (SA) based SOS (SASOS) in order to improve the convergence rate and quality of solution of SOS. The SOS algorithm has a strong global exploration capability and uses fewer parameters. The systematic reasoning ability of SA is employed to find better solutions on local solution regions, hence, adding exploration ability to SOS. Also, a fitness function was proposed which takes into account the utilization level of virtual machines (VMs) which reduced make span and degree of imbalance among VMs. CloudSim tool kit was used to evaluate the efficiency of the proposed method using both synthetic and standard workloads. Results of simulation showed that hybrid SOS performs better than SOS in terms of convergence speed, response time, degree of imbalance, and makespan.

Maguluri *et al.* [28] presented a stochastic model of load balancing and scheduling in cloud computing clusters, it shows that the widely-used best fit scheduling algorithm is not throughput-optimal, and present alternatives that achieve any arbitrary fraction of the capacity region of the cloud and also study the delay performance of these alternative algorithms through simulations. Li *et al.* [29] constructed an improved first in-first out (FIFO) scheduling algorithm based on fuzzy clustering in cloud computing to reduce tasks' waiting time. The researcher constructs a task model, resource model, and analyzes tasks' preference, then classifies resources with fuzzy clustering algorithms. As a result, the algorithm reduced tasks' waiting time and improved resource utilization. The experiment results showed that the proposed algorithm shortens the execution time of tasks and increases resource utilization [30]–[32].

Maurya *et al.* [33] proposes a multi-objective load balancing in cloud computing using a meta-heuristic approach. The suggested multi-objective load balancing paradigm considers power usage, bandwidth consumption, migration costs, memory usage, and load balancing parameters (response time, turnaround time, server load). A novel hybrid optimization technique-mouse customized golden eagle optimization (MCGEO) model is developed for optimal load balancing, which is conceptual combination of traditional golden eagle optimizer (GEO) and cat and mouse-based optimizer (CMBO). Therefore, the newly developed hybrid optimization model improves convergence and addresses the optimization problems in load balancing. The proposed MCGEO achieved a throughput value of ∼281.6255, at 150 tasks for the cloud environment-2 this proves the superiority of the proposed approach. Al-Fatlawi and Al-Barazanchi [34] proposed a hybrid scheduling policy which is hybrid of both particle swarm optimization (PSO) algorithm and actor-critic algorithm named as hybrid particle swarm optimization actor critic (HPSOAC) to solve this issue. This hybrid scheduling policy helps each agent to improve individual learning as well as learning through exchanging information among other agents. An experiment is carried out with the help of Python simulator with TensorFlow. Outcome shows that our proposed scheduling policy reduces 5.16% and 10.86% in energy consumption, reduces 7.13% and 10.04% in makespan time, and has marginally better resource utilization over deep Q-network (DQN) and Q-learning based on modified particle swarm optimization (QMPSO) algorithm, respectively. In [35] developed a bio inspired improved lion optimization meta-heuristic approach for solving the load Balancing issues in the cloud. This Meta-Heuristic approach has better exploration and exploitation rate when compared with other bio inspired algorithms and it does not get struck into local optima during the search process of identifying the underutilized node. This proposed work has been implemented in CloudSim and its performance is going to be evaluated against the benchmarks that are identified from the literature

Husain *et al.* [9] developed an algorithm for controlling job distribution at the database servers in different node partitions. The load status of the database servers depends on three important parameters namely processor, RAM, and bandwidth. On the basis of load status, the clients'/users' requests can then be directed to another database in a distributed environment in order to balance the load effectively to meet user demands in an unobtrusive manner. Arising from the review of related works, it can be observed that there is a need for efficient task scheduling or load balancing algorithms for optimum resource utilization, waiting time, evading of application unresponsiveness and non–dynamic partitioning of resources in the cloud computing environment.

Arising from Table 5 and Figure 4, it can be observed that dynamic technique for load balancing in cloud computing seems to be the most adopted technique by most researchers and also, that dynamic handles most of the evaluation metrics well compared to static techniques [36]. Also, there is still need for efficient task scheduling or load balancing techniques for waiting time, a good quality of service (QoS) system, evade of application unresponsiveness and non–dynamic partitioning of resources in the cloud computing environment [37]–[40]. It can be deduced that load balancing is a major issue in cloud computing which caught the interest of research and there still exist a lot of research gaps that need to be filled in load balancing to achieve a robust and satisfactory computing system.

Table 5. Summary of reviewed technique/algorithm, software used and performance metrics

Authors and years	Technique/algorithm	Software used	Performance metrics	Scopes not covered by the author
[23]	Dynamic Round Robin	CloudSim	Response time and bandwidth	Network performance
[3]	Round Robin	VM	Response and execution time	Cloud division technique
[41]	GA	CloudAnalyst	Response time	QoS
[42]	GA	CloudSim	Makespan	Utilizes the resource efficiently and ensures QoS.
[25]	ACO and HoneyBee	VM	Response time	Non-dynamic partitioning of resource
[43]	Developed an algorithm	CloudSim	Response time	Other performance metrics not considered
[27]	GA	CloudAnalyst	Makespan	Network performance
[5]	ACO	CloudSim	Execution time and degree of load balancing	Overall system performance and efficiency
[44]	MDAPSO	Cloudlet	Execution time and resource utilization	Software complexity
[7]	SASOS	CloudSim	Response time, degree of imbalance, makespan and convergence speed	Effective local search
[28]	Best – fit Scheduling algorithm	CloudSim	Delay in time	Throughput was not considered
[29]	FIFO	CloudSim	Execution time, resource utilization	QoS was not covered
[9]	Node partitioning algorithm	CloudSim	Response time	QoS
[45]	ETVMC	CloudSim	Time consumption, energy consumption, task rejection and makespan	Focused was limited on energy efficiency
[46]	Cloud service agent	CloudSim	Throughput, obtained time	QoS not covered
[47]	Zero imbalance approach	VM	Makespan and resource utilization	Power consumption and VM live migration
[48]	BWM and VIKOR	VM	Throughput, makespan, waiting time and resource utilization	QoS not covered
[49]	MCGEO	CloudSim	Resource time, turnaround time and server load	QoS was not considered
[50]	HPSOAC	TensorFlow	Energy, makespan and resource utilization	Throughput was not considered
[51]	Lion optimization algorithm	CloudSim	Throughput, migration and response time	QoS and energy consumption not covered

Note: MDAPSO: modified dynamic adaptive particle swarm optimization

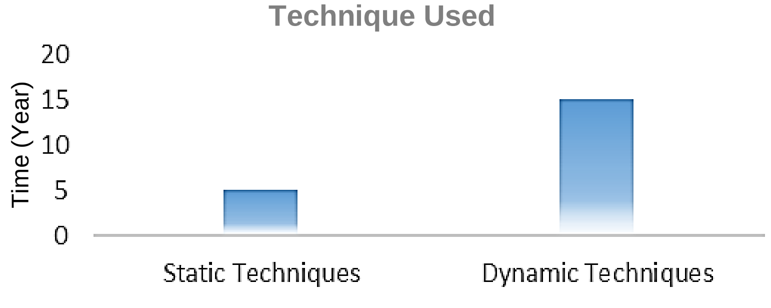


Figure 4. Chart showing the type of techniques used by the reviewed article

5. CONCLUSION

This study provides an in-depth examination of load balancing in cloud computing. The paper goes over an overview of cloud computing, its potentials, load balancing in cloud computing, challenges in load balancing, load balancing techniques in cloud computing, algorithms used for load balancing in cloud computing, and evaluation metrics for load balancing in cloud computing in detail, also present the recent advancement in load balancing techniques in cloud computing systems. The deployment practicality of load balancing in cloud computing has been thoroughly analyzed, and its application has been assessed using a range of metrics to measure performance in order to produce a stable and efficient system. It is worth noting, however, that in the design of an effective cloud computing system, the crucial theme to keep in mind at the planning stage is the pre-determination of load balancing approach. Despite the fact that many authors employed optimization techniques, it is clear that a robust algorithm/approach for load balancing that takes into account QoS and dynamic loading of workloads in cloud computing is required.

REFERENCES

[1] A. Heidari, N. J. Navimipour, M. A. J. Jamali, and S. Akbarpour, "A hybrid approach for latency and battery lifetime optimization in IoT devices through offloading and CNN learning," *Sustainable Computing: Informatics and Systems*, vol. 39, Sep. 2023, doi: 10.1016/j.suscom.2023.100899.

[2] S. O. Olabiyisi, T. M. Fagbola, and R. S. Babatunde, "An exploratory study of cloud and ubiquitous computing systems," *World Journal of Engineering and Pure and Applied Sciences*, vol. 2, no. 5, pp. 148–154, 2012.

[3] A. Ajil and E. S. Kumar, "A comprehensive study of LB technique in cloud infrastructure," *SN Computer Science*, vol. 4, no. 2, Jan. 2023, doi: 10.1007/s42979-022-01588-x.

[4] R. Rawat, O. A. Oki, K. S. Sankaran, O. Olasupo, G. N. Ebong, and S. A. Ajagbe, "A new solution for cyber security in big data using machine learning approach," in *Mobile Computing and Sustainable Informatics: Proceedings of ICMCSI 2023*, 2023, pp. 495–505, doi: 10.1007/978-981-99-0835-6_35.

[5] R. Gao and J. Wu, "Dynamic load balancing strategy for cloud computing with ant colony optimization," *Future Internet*, vol. 7, no. 4, pp. 465–483, Nov. 2015, doi: 10.3390/fi7040465.

[6] I. Aqeel *et al.*, "Load balancing using artificial intelligence for cloud-enabled internet of everything in healthcare domain," *Sensors*, vol. 23, no. 11, Jun. 2023, doi: 10.3390/s23115349.

[7] M. Abdullahi and M. A. Ngadi, "Hybrid symbiotic organisms search optimization algorithm for scheduling of tasks on cloud computing environment," *PLOS ONE*, vol. 11, no. 6, Jun. 2016, doi: 10.1371/journal.pone.0158229.

[8] R. Bi, T. Peng, J. Ren, X. Fang, and G. Tan, "Joint service placement and computation scheduling in edge clouds," in *2022 IEEE International Conference on Web Services (ICWS)*, Jul. 2022, pp. 47–56, doi: 10.1109/ICWS55610.2022.00022.

[9] D. Husain, M. Omar, K. Ahmad, V. Jain, and R. Wason, "Load status evaluation for load balancing in distributed database servers," *3C Tecnología_Glosas de innovación aplicadas a la pyme*, pp. 422–447, May 2019, doi: 10.17993/3ctecno.2019.specialissue2.422-447.

[10] A. M. Algashami, "Algorithms for the executable programs planning on supercomputers," *PLOS ONE*, vol. 17, no. 9, Sep. 2022, doi: 10.1371/journal.pone.0275099.

[11] S. A. Ajagbe, M. O. Ayegboyin, I. R. Idowu, T. A. Adeleke, and D. N. H. Thanh, "Investigating energy efficiency of mobile ad-hoc network routing protocols," *Informatica*, vol. 46, no. 2, Jun. 2022, doi: 10.31449/inf.v46i2.3576.

[12] A. Ullah, N. M. Nawi, and S. Ouhame, "Recent advancement in VM task allocation system for cloud computing: review from 2015 to 2021," *Artificial Intelligence Review*, vol. 55, no. 3, pp. 2529–2573, Mar. 2022, doi: 10.1007/s10462-021-10071-7.

[13] A. Sanjay, P. V. V. Sai, and G. Ar, "Energy efficient resource load balancing in cloud computing," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 7, pp. 704–707, 2019.

[14] S. A. Ajagbe, M. O. Oyediran, A. Nayyar, J. A. Awokola, and J. F. Al-Amri, "P-acohoneybee: a novel load balancer for cloud computing using mathematical approach," *Computers, Materials & Continua*, vol. 73, no. 1, pp. 1943–1959, 2022, doi: 10.32604/cmc.2022.028331.

[15] V. K. Naik, K. Beaty, and A. Kundu, "Service usage metering in hybrid cloud environments," in *2014 IEEE International Conference on Cloud Engineering*, Mar. 2014, pp. 253–260, doi: 10.1109/IC2E.2014.76.

[16] S. G. Domanal and G. R. M. Reddy, "Optimal load balancing in cloud computing by efficient utilization of virtual machines," in *2014 Sixth International Conference on Communication Systems and Networks (COMSNETS)*, Jan. 2014, pp. 1–4, doi: 10.1109/COMSNETS.2014.6734930.

[17] P. M. Shameem and R. S. Shaji, "A methodological survey on load balancing techniques in cloud computing," *International Journal of Engineering and Technology (IJET)*, vol. 5, no. 5, pp. 3801–3812, 2013.

[18] Z. Wang, Y. Zhao, H. Yong, Z. Liu, W. Wang, and Y. Lu, "The contribution of gut bacteria to pesticide resistance of Tribolium castaneum (Herbst)," *Journal of Stored Products Research*, vol. 103, Sep. 2023, doi: 10.1016/j.jspr.2023.102160.

[19] G. Zhang, L. Liu, and H. Guo, "Investigating the impact of cloud computing vendor on the adoption of cloud computing," *Mobile Information Systems*, vol. 2021, pp. 1–18, Dec. 2021, doi: 10.1155/2021/6557937.

[20] A. A. A. Alkhatib, A. Alsabbagh, R. Maraqa, and S. Alzubi, "Load balancing techniques in cloud computing: Extensive review," *Advances in Science, Technology and Engineering Systems Journal*, vol. 6, no. 2, pp. 860–870, Apr. 2021, doi: 10.25046/aj060299.

[21] V. Ramaswamy, *Co-creation of value-towards an expanded paradigm of value creation*. Marketing Review St. Gallen, 2009.

[22] J. B. Awotunde *et al.*, "Cyber-physical systems security: analysis, opportunities, challenges, and future prospects," in *Blockchain for Cybersecurity in Cyber-Physical Systems. Advances in Information Security*, 2023, pp. 21–46.

[23] T. Samson Adekunle, M. Oladayo Lawrence, O. Omotayo Alabi, A. A. Afolorunso, G. Nse Ebong, and M. Abiola Oladipupo, "Deep learning technique for plant disease detection," *Computer Science and Information Technologies*, vol. 5, no. 1, pp. 49–56, Mar. 2024, doi: 10.11591/csit.v5i1.p49-56.

[24] A. B. Kanbar and K. Faraj, "Modern load balancing techniques and their effects on cloud computing," *Journal of Hunan University Natural Sciences*, vol. 49, no. 7, pp. 37–43, Jul. 2022, doi: 10.55463/issn.1674-2974.49.7.5.

[25] A. Nadaph and V. Maral, "Cloud computing - partitioning algorithm and load balancing algorithm," *International Journal of Computer Science, Engineering and Information Technology*, vol. 4, no. 5, pp. 41–49, Oct. 2014, doi: 10.5121/ijcseit.2014.4504.

[26] Z. Xiao, M. Wang, A. T. Chronopoulos, and J. Jiang, "A method for reducing cloud service request peaks based on game theory," *Journal of Parallel and Distributed Computing*, vol. 165, pp. 107–119, 2022.

[27] P.-H. Liang and J.-M. Yang, "Evaluation of two-level global load balancing framework in cloud environment," *International Journal of Computer Science and Information Technology*, vol. 7, no. 2, pp. 1–11, Apr. 2015, doi: 10.5121/ijcsit.2015.7201.

[28] S. T. Maguluri, R. Srikant, and L. Ying, "Stochastic models of load balancing and scheduling in cloud computing clusters," in *2012 Proceedings IEEE INFOCOM*, Mar. 2012, pp. 702–710, doi: 10.1109/INFCOM.2012.6195815.

[29] J. Li, T. Ma, M. Tang, W. Shen, and Y. Jin, "Improved FIFO scheduling algorithm based on fuzzy clustering in cloud

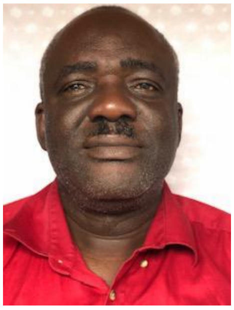
computing," *Information*, vol. 8, no. 1, Feb. 2017, doi: 10.3390/info8010025.
[30] N. G. Puttaswamy and A. N. Murthy, "An efficient reconfigurable workload balancing scheme for fog computing network using internet of things devices," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 6, pp. 6525–6537, Dec. 2023, doi: 10.11591/ijece.v13i6.pp6525-6537.
[31] A. Anjum and A. Parveen, "Optimized load balancing mechanism in parallel computing for workflow in cloud computing environment," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 12, no. 2, pp. 276–286, Jul. 2023, doi: 10.11591/ijres.v12.i2.pp276-286.
[32] N. George, A. B. Kadan, and V. P. Vijayan, "Multi-objective load balancing in cloud infrastructure through fuzzy based decision making and genetic algorithm based optimization," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 12, no. 2, pp. 678–685, Jun. 2023, doi: 10.11591/ijai.v12.i2.pp678-685.
[33] S. K. Maurya, S. Malik, and N. Kumar, "Virtual machine tree task scheduling for load balancing in cloud computing," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 30, no. 1, pp. 388-393, Jul. 2023, doi: 10.11591/ijeecs.v30.i1.pp388-393
[34] A. A. Al-Fatlawi and I. Al-Barazanchi, "A novel approach for new architecture for green data centre," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 12, no. 1, pp. 411–417, Feb. 2023, doi: 10.11591/eei.v12i1.4421.
[35] R. Ravinder and N. Vurukonda, "Cloud computing: an efficient load balancing and scheduling of task method using a hybrid optimization algorithm," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 32, no. 3, pp. 1545–1556, Dec. 2023, doi: 10.11591/ijeecs.v32.i3.pp1545-1556.
[36] A. S. Chandrashekhar and S. Gunakimath Suryakanth, "A combined computing framework for load balancing in multi-tenant cloud eco-system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 5, pp. 5630–5643, Oct. 2022, doi: 10.11591/ijece.v12i5.pp5630-5643.
[37] A. S. Joshi and S. Devi Munisamy, "In-depth analysis of dynamic degree load balancing technique in public cloud for heterogeneous cloudlets," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 27, no. 2, pp. 1119–1126, Aug. 2022, doi: 10.11591/ijeecs.v27.i2.pp1119-1126.
[38] M. M. Akawee, M. A. Ahmed, and R. A. Hasan, "Using resource allocation for seamless service provisioning in cloud computing," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 26, no. 2, pp. 854–858, May 2022, doi: 10.11591/ijeecs.v26.i2.pp854-858.
[39] N. G. Elnagar, G. F. Elkabbany, A. A. Al-Awamry, and M. B. Abdelhalim, "Simulation and performance assessment of a modified throttled load balancing algorithm in cloud computing environment," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, pp. 2087–2096, Apr. 2022, doi: 10.11591/ijece.v12i2.pp2087-2096.
[40] A. Saoud and A. Recioui, "Hybrid algorithm for cloud-fog system based load balancing in smart grids," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 11, no. 1, pp. 477–487, Feb. 2022, doi: 10.11591/eei.v11i1.3450.
[41] S. Afzal and G. Kavitha, "Load balancing in cloud computing – a hierarchical taxonomical classification," *Journal of Cloud Computing*, vol. 8, no. 1, Dec. 2019, doi: 10.1186/s13677-019-0146-7.
[42] H. Luo, C. Yan, and J. Luo, "Dynamic programming based grid workflow scheduling algorithm," in *Software Engineering and Knowledge Engineering: Theory and Practice*, 2012, pp. 993–1000.
[43] W. Saber, W. Moussa, A. M. Ghuniem, and R. Rizk, "Hybrid load balance based on genetic algorithm in cloud environment," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 3, pp. 2477–2489, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2477-2489.
[44] A. Al-maamari and F. A. Omara, "Task scheduling using PSO algorithm in cloud computing environments," *International Journal of Grid and Distributed Computing*, vol. 8, no. 5, pp. 245–256, Oct. 2015, doi: 10.14257/ijgdc.2015.8.5.24.
[45] G. Singh, S. Kumar, and S. Prakash, "A performance improvement model for cloud computing using simulated annealing algorithm," *International Journal of Software Innovation*, vol. 10, no. 1, pp. 1–17, May 2022, doi: 10.4018/IJSI.301222.
[46] L. Kong, J. P. B. Mapetu, and Z. Chen, "Heuristic load balancing based zero imbalance mechanism in cloud computing," *Journal of Grid Computing*, vol. 18, no. 1, pp. 123–148, Mar. 2020, doi: 10.1007/s10723-019-09486-y.
[47] E. Rafieyan, R. Khorsand, and M. Ramezanpour, "An adaptive scheduling approach based on integrated best-worst and VIKOR for cloud computing," *Computers & Industrial Engineering*, vol. 140, Feb. 2020, doi: 10.1016/j.cie.2020.106272.
[48] K. V. Kumar and A. Rajesh, "Multi-objective load balancing in cloud computing: a meta-heuristic approach," *Cybernetics and Systems*, vol. 54, no. 8, pp. 1466–1493, Nov. 2023, doi: 10.1080/01969722.2022.2145656.
[49] A. Pradhan, S. K. Bisoy, and M. Sain, "Action-based load balancing technique in cloud network using actor-critic-swarm optimization," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–17, Jun. 2022, doi: 10.1155/2022/6456242.
[50] R. Kaviarasan, G. Balamurugan, R. Kalaiyarasan, and Y. V. R. Reddy, "Effective load balancing approach in cloud computing using inspired lion optimization algorithm," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 6, no. December, Dec. 2023, doi: 10.1016/j.prime.2023.100326.
[51] A. S. Kadhim and M. E. Manaa, "Hybrid load-balancing algorithm for distributed fog computing in internet of things environment," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 11, no. 6, pp. 3462–3470, Dec. 2022, doi: 10.11591/eei.v11i6.4127.

BIOGRAPHIES OF AUTHORS

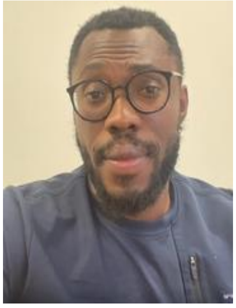
Mayowa O. Oyediran is a senior lecturer in the Department of Computer Engineering in Ajayi Crowther University, Oyo town in Nigeria. He has a PhD in computer engineering from the Ladoke Akintola University of Technology, Ogbomoso, Nigeria. He specializes in distributed systems and applications with focused on cloud computing, fog computing, MANETs and the likes. He can be contacted at email: mo.oyediran@acu.edu.ng and his profile can be found at https://www.researchgate.net/profile/Oyediran-Mayowa.

Olufemi S. Ojo is currently a lecturer at the Department of Computer Sciences, Ajayi Crowther University Oyo town in Nigeria, He has a Ph.D in computer science from Ladoke Akintola University of Technology, Ogbomoso. Nigeria. His research focuses on pattern recognition, optimization techniques and e-learning. He is a full member of the International Association of Engineers (IAEng) and Computer Professionals of Nigeria (CPN). He can be contacted at email: os.ojo@acu.edu.ng and his profile can be found at https://www.researchgate.net/profile/Olufemi-Ojo-2.

Sunday Adeola Ajagbe is a Ph.D candidate at the Department of Computer Science, University of Zululand, South Africa and a lecturer, a First Technical University, Ibadan, Nigeria. He obtained MPhil computer engineering at Ladoke Akintola University of Technology (LAUTECH), Ogbomoso, MSc and BSc in information technology and communication technology respectively at the National Open University of Nigeria (NOUN), and his Postgraduate Diploma in electronics and electrical engineering at LAUTECH. His specialization includes artificial intelligence (AI), natural language processing, information security, data science, and the internet of things (IoT). He is also licensed by The Council Regulating Engineering in Nigeria as a professional Electrical Engineer, and International Association of Engineers (IAENG). He has many publications to his credit in reputable academic databases. He can be contacted at email: saajagbe@pgschool.lautech.edu.ng and his profile can be found at https://www.researchgate.net/profile/Sunday-Ajagbe.



Olukayode Aiyeniko received B.Eng. degree in mechanical engineering from Ahmadu Bello university, Zaria, Nigeria; PGD and MSc degrees in computer science from University of Lagos, Nigeria. PhD in computer science from Babcock University, Ogun state, Nigeria. Post graduate program certificate in data science and business analytics as well as in artificial intelligence from University of Texas, Austin, USA. His major research interests are in artificial intelligence and information security. He can be contacted at email: olukayode.aiyeniko@lasu.edu.ng and his profile can be found at https://www.researchgate.net/profile/Olukayode-Aiyeniko.



Princewill Chima Obuzor is a Ph.D. Candidate in data mining at the University of Salford in Manchester United Kingdom. He obtained a Masters in databases and web-based systems at the University of Salford and a B.Sc. with Distinction at V. Dahl East Ukrainian National University in Luhansk, Ukraine. His research interests are in machine learning and artificial intelligence, data science, process mining and blockchain technologies. He is a member of the British Computer Society (BCS) in the UK. He can be contacted at email: p.obuzor@edu.salford.ac.uk, and his profile can be found at https://www.researchgate.net/profile/Princewill-Obuzor.

Matthew Olusegun Adigun retired in 2020 as a senior professor of computer science at the University of Zululand. He obtained his doctorate degree in 1989 from Obafemi Awolowo University, Nigeria; having previously received both master's in computer science (1984) and a Combined Honours degree in computer science and economics (1979) from the same University (when it was known as University of Ife, Nigeria). He has received both research and teaching recognitions for raising the flag of excellence in historically disadvantaged South African Universities as well as being awarded a 2020 SAICSIT Pioneer of the Year in the computing discipline. Currently, he works as a Temporary Senior Professor at the Department of Computer Science, University of Zululand, Kwadlangezwa, South Africa to pursue his recent interest in AI-enabled pandemic response and preparedness. He can be contacted at email: adigunm@unizulu.ac.za, and his profile can be found at https://www.researchgate.net/profile/Matthew-Adigun.