OPEN ACCESS

ISSN: 2996-4954

Prasanth K, et al.
J Contemp Edu Theo Artific Intel, 2023; 4(1):100105
DOI: 10.71010/2996-4954/JCETAI-105

Research Article

Journal of Contemporary Education Theory & Artificial Intelligence

Allocating Resources in Cloud Computing: An Examination of Scheduling Algorithms and Optimisation Methods

Prasanth Kosaraju[1*], Venu Madhav Nadella[2], Sanketh Nelavelli[3], Ramadevi Sannapureddy[4]

[1]Dataquest Corp, Network Engineer,kosarajup72@gmail.com
[2]Cyma systems inc, Technical Analyst IV, reachvenunadella@gmail.com
[3]MSR technologies, Cloud Engineer, sanketh.nsk@gmail.com
[4]SealedAir Corporation, Sr. IT Systems Engineer, ramadevi.sannapureddy@gmail.com

[*]**Corresponding author:** Prasanth Kosaraju

IJCACI and Washington University of Science and Technology (WUST) Conference Proceedings 2025
https://scrs.in/conference/ijcaci2025/page/Best%20Paper%20Award

Citation: Prasanth K, Venu Madhav N, Sanketh N, Ramadevi S (2023) Allocating Resources in Cloud Computing: An Examination of Scheduling Algorithms and Optimisation Methods. J Contemp Edu Theo Artific Intel: JCETAI-105.

Received Date: 05 Sep, 2023; **Accepted Date:** 17 Sep, 2023; **Published Date:** 25 Sep, 2023

Abstract

Cloud computing has revolutionized large-scale computing and data storage by providing on-demand access to resources located anywhere in the world. To achieve peak performance, dependable service, and cost-effectiveness in cloud settings, efficient allocation of resources is essential. In this paper, it will go over the basics of cloud computing, including how it works, how to schedule tasks, and how to optimize resource allocation. It examines centralized vs. decentralized systems, discusses static and dynamic scheduling strategies, and rates popular algorithms including FCFS, Round Robin, Min-Min, and Max-Min. Key enablers of intelligent and adaptive resource allocation are heuristic, metaheuristic, and AI-driven methodologies, such as machine learning and deep learning techniques, according to the report. The present status of resource management in cloud systems can be better understood with the help of this all-encompassing review, which also points the way toward potential avenues for further study and improvement.

Keywords: *Cloud Computing, Resource Allocation, Task Scheduling, Optimization Techniques, Static and Dynamic Scheduling, Centralized and Decentralized Scheduling, Metaheuristics, Performance Efficiency*.

I. Introduction

Cloud computing has become increasingly popular for large-scale computation and data storage due to its ability to provide shared access to distributed computing resources worldwide. Cloud infrastructures eliminate the physical limitations of traditional isolated systems, enabling the dynamic management of resources as unified entities [1]. People using their own devices and large corporations operating hundreds of virtual machines in data centres across the world are both served by these platforms, which provide storage and compute capacity on a scalable basis. Cloud computing represents a natural progression in data centre evolution, driven by automation, workload balancing, and virtualization technologies [2]. It integrates geographically dispersed resources into a unified, on-demand computing environment.

Computational resources like CPU, memory, and storage must be allocated efficiently; this is one of the major obstacles in cloud computing [3]. Problems with resource management persist because user needs are both varied and ever-changing. Cloud providers are tasked with meeting diverse Quality of Service (QoS) requirements while ensuring fair and efficient resource utilization. However, limited on-site resources and unpredictable workloads make it difficult to meet all service requests promptly and consistently.

The ability to rent virtualized computer resources in the cloud allows for scalable and adaptable service provisioning [4]. While this virtual infrastructure supports on-demand provisioning, rapid fluctuations in workloads can degrade service quality [5]. To address this, cloud service providers (CSPs) must implement scalable and resilient resource management strategies that dynamically reconfigure both hardware and software components in response to variations in demand.

ML methods have lately become effective resources for solving cloud computing resource allocation problems. ML algorithms are capable of automating decision-making and learning optimal or near-optimal lively infer patterns, even in large-scale, heterogeneous contexts [6]. However, conventional ML techniques may struggle with high-dimensional data, non-uniform formats, and complex interdependencies commonly seen in cloud data centres. These limitations necessitate manual feature engineering to ensure accurate modelling. In contrast, deep learning (DL), a subset of ML, offers a more robust approach by automatically extracting hierarchical features through layered neural architectures.

A. Structure of the paper

This paper is structured in the following way: The II section delves into the basics and difficulties of cloud resource allocation. A cloud scheduling method classification is provided in Section III. Methods for scheduling and optimization that are often employed are reviewed in Section IV. A review of the relevant literature is presented in Section V. Section VI provides a summary of the paper's main points and a roadmap for further study.

II. Fundamentals of Resource Allocation in Cloud Environments

In the cloud, resource allocation is a way to make sure that apps and virtual machines (VMs) get the most out of the computing resources that are available. One of the focuses considered in the cloud environment that has implications on the efficiency of the system attained, user satisfaction with how the cloud performs its service, and the cost efficiency of delivering the service, is

resource allocation [7]. Cloud environments are a heterogeneous and dynamic environment, and therefore, they must be in a position to allocate and de-allocate resources in real-time to various workloads. Some of the issues of resource allocation are important, and they are as follows: workload characterization, task scheduling, and resource provisioning. Besides characterization of workloads and energy efficiency, service level agreements (SLAs), and cost, resource allocation mechanisms must consider scalability [8]. Cloud environments depend considerably on Virtualization, which is the encapsulation abstraction that empowers the flexile (and worth) of a physical resource.

B. Key Concepts of Resurce Allocation in Cloud (Virtualization, Multi-Tenancy, Elasticity)

Key enabling technologies include virtualization, which abstracts physical hardware to allow multiple virtual instances, and multi-tenancy, which supports concurrent use of infrastructure by different users while maintaining isolation. Elasticity is another core concept, allowing resources to scale up or down dynamically in response to workload variations.

1) Virtualization

Virtualization" refers to the process of making an imitation of an object or system rather than the real thing itself. A key part of cloud computing, virtualisation creates digital copies of physical resources so that resources can be used more effectively [9]. An actual computer can be used to host VMs, which share resources like memory, storage, processing power, and network speed. Each VM can run its own operating system and set of programs [10]. This facilitates resource sharing among multiple users or tenants while maintaining isolation and operational flexibility.

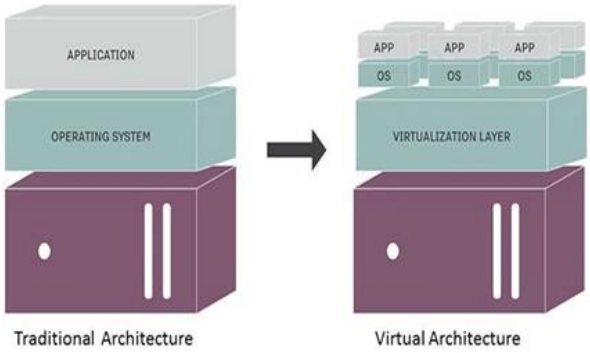

Fig. 1. Traditional Architecture and Virtual Architecture.

Virtualization improves infrastructure utilization, reduces cost, and enhances the agility of IT processes. It enables cloud providers to meet dynamic user demands without the need for heavy investment in physical infrastructure. As illustrated in Figure 1, virtualization shifts the architecture from a traditional model where resources are tightly coupled with hardware to a virtualized model that supports multiple VMs on a single host machine. Virtualization in its many variants helps create efficient and scalable cloud environments; it is used for storage, networks, desktops, and appliances, among others.

2) Multi-Tenancy

Cloud computing's "multi-tenancy" describes how various users or tenants share the same platform's storage, processing power, and network infrastructure [11]. This model increases resource utilization and cost-efficiency by allowing a single physical infrastructure to support multiple clients simultaneously. However, it also introduces critical security and privacy challenges. Shared resource objects must be carefully managed, as they can become potential points of data leakage or confidentiality breaches. While cloud environments provide strong separation at the application and virtual machine levels, isolation at the hardware level IOT is often limited. One particular risk is data remanence, where residual data remains on storage media after deallocation, potentially allowing a malicious tenant to recover sensitive information from previously used resources. Therefore, effective isolation, secure deletion, and resource sanitization mechanisms are essential to maintain trust and ensure data privacy in multi-tenant cloud infrastructures.

3) Elasticity

The ability for applications and services to dynamically scale up or down in response to fluctuating workload needs is a key characteristic of cloud computing [12]. Automated acquisition and release of shared computing resources, such as processing power, memory, and storage, via cloud systems during runtime optimises performance and reduces costs. This capability is central to the economies of scale offered by cloud platforms and enhances the utility and flexibility of cloud services. The NIST definition emphasizes elasticity as the rapid provisioning and de-provisioning of virtually unlimited resources, available for purchase on demand. From the cloud consumer's perspective, elasticity is assessed based on how quickly and precisely the platform can respond to workload fluctuations. Effective elasticity ensures that applications maintain service quality during peak loads while minimizing cost during idle periods.

4) Resource Types

Cloud computing resource management is based on four main types of resources: CPU, memory, storage, and network bandwidth. Execution of programs and other duties is defined by the Central Processing Unit (CPU). The RAM is the short-term storage space that programs use to process data; if there isn't enough RAM, the system could crash or run slowly. Storage is the long-term data storage area in which files, databases, and system images are stored; it should be provisioned both capacity-wise and performance-wise. The bandwidth of the network is autocratic by the speed at which data is moved among the parts of the cloud and the users; insufficient bandwidth may create latency and bottlenecks, and poor performance of applications. Such a task as balancing these resources according to the peculiarities of the workloads is key to optimal cloud performance, cost-efficiency, and scalability.

C. Challenges in Resource Allocation in Cloud Computing

Cloud computing still has a lot of problems with allocating resources. This is because end users can access resources from anywhere at any time, and application tasks are always changing. Resources are accessed indirectly via SOAP or RESTful APIs, which map user requests to virtualized components such as servers, storage, or elastic Ips [13]. While cloud data centres offer a vast pool of resources to support on-demand elasticity, this abundance can lead to non-optimal allocation if not managed effectively.

According to their findings, energy-efficient resource allocation policies face a number of obstacles, including:

- Factors including resource utilization, performance, and power consumption, as well as the type of job and any conflict between them, must be carefully considered.
- Allocating and using resources in real time by investigating the potential for unified, shared, and standardized datacentre resources.
- Faster recovery from failures while enhancing asset usage, network accessibility, and power efficiency.

- Analysis and optimization of the cloud infrastructure design for better use of cloud resources, architecture, tools, and technology.
- Analyzing application interdependencies to enable resource consolidation, which increases performance and ROI.
- Facilitating adaptability and safety of mission-critical applications by way of realistic cloud infrastructure planning.

III. Classification and Commonly Used Scheduling Algorithms in Cloud Computing

Resource scheduling is a key part of cloud computing that makes sure resources are used well and jobs are finished on time. It does this by deciding how and when real and logical resources are assigned to users. While resource provisioning involves the identification, selection, and allocation of necessary resources to execute workflows [14], resource scheduling focuses on creating a schedule that maps tasks to appropriate resources based on various criteria. Effective scheduling not only maximizes resource usage but also ensures compliance with SLAs and QoS requirements. This paper focuses specifically on the scheduling phase, exploring commonly used algorithms and approaches designed to produce efficient task-resource mappings in cloud environments.

D. Static vs. Dynamic Scheduling

Static scheduling pre-defines task allocation before execution based on complete knowledge of resources, making it suitable for stable environments. In contrast, dynamic scheduling adapts in real-time to workload changes, providing better flexibility and efficiency in heterogeneous cloud systems.

1) Static Scheduling

In cloud computing, static task scheduling algorithms, resource allocation, and task execution routines are set up ahead of time, assuming that all tasks and system resources are known in full. These algorithms are adapted to a homogeneous and predictable cloud environment because they have low scheduling overhead. With regard to performance, under stable workloads, static scheduling has the potential to lead to high resource utilization, smaller make span, as well as better throughput. But they fail in dynamic cases as they are not flexible and cannot adjust to the dynamism in the tasks or the resource availability [15]. The variable conditions can result in worse response time and, therefore, cannot be used in real-time or large-scale cloud environments. What is more, a significant number of static scheduling methods are highly dependent on simulated or virtualized environments as opposed to actual cloud environments, and thus may lead to a gap between theoretical performance and practical deployment performance. Thus, although the computation time of static algorithms is efficient, their use in practice in contemporary, heterogeneous cloud environments is restricted.

2) Dynamic Scheduling

The Mobility Directed approach is comparable to this approach. At each scheduling phase, it assigns an attribute called dynamic level (DL) to non-planned nodes, allowing it to dynamically set their priorities [16]. What follows is a detailed explanation of the DLS:

- A static b-level is computed at each node.
- The admitted nodes are the sole members of the initial group of prepared.
- Every node's EBT is computed at its processor.
- Prioritize the node processor pairs that produce the highest DL. Scheduled to the same processor are all nodes.
- The final phase is to schedule each node by adding it to the prepared group.

E. Centralized Vs. Decentralized Scheduling

Centralized scheduling relies on a single controller for resource allocation, offering simplicity, efficient management, and enhanced control over resources [17]. However, it lacks scalability and fault tolerance, making it unsuitable for large-scale and dynamic environments. On the other hand, decentralized scheduling distributes decision-making across multiple local schedulers or nodes [18]. This approach improves scalability, adaptability, and fault tolerance, particularly in heterogeneous and large-scale cloud systems, though it may lead to sub-optimal resource utilization and potential load imbalance due to the lack of a global view. The following Table I: provides a detailed comparison between centralized and decentralized scheduling:

Table 1: Centralized vs Decentralized with aspects.

Aspect	Centralized Scheduling	Decentralized Scheduling
Definition	A single scheduler makes global decisions for resource allocation.	Multiple local schedulers make independent decisions with limited or local information.
Advantages	Easy to manage- High efficiency- Better resource control and monitoring	Highly scalable- Fault tolerant- More adaptable to changes
Disadvantages	Poor scalability- Limited fault tolerance- May underperform in large-scale grids	May lead to sub-optimal resource use- Potential for load imbalance
Scalability	Limited	High
Fault Tolerance	Low	High
Suitability	Suitable for small-scale systems	More suitable for dynamic and large-scale distributed systems
Implementation Complexity	Simpler to implement	More complex due to the need for coordination

This comparison highlights the trade-offs between centralized and decentralized approaches, helping guide the choice based on the system's size, complexity, and operational requirements.

F. Common Scheduling Algorithms

Optimising performance, boosting resource utilisation, and assuring timely execution are all greatly impacted by appropriate task scheduling in cloud computing systems. Several scheduling algorithms have been developed to address different workload patterns, system architectures, and performance requirements. Some of the most commonly used scheduling algorithms are discussed below, including FCFS, RR, Min-Min, Max-Min, GA, ACO, and Particle Swarm Optimization (PSO). Each algorithm has its own strengths and limitations depending on the nature of the tasks and system conditions.

1) First-Come-First-Served (FCFS)

The simplest and fastest approach is to prioritize the execution of the first submitted process. Scheduling in this manner is known as FCFS. A process is effectively moved to the very end of a queue the moment it is submitted. Whenever a process completes its execution, the next one is retrieved from the top of the queue. Figure. 2, with its four-state graphic, illustrates this point. FCFS is fair and non-pre-emptive, although it may cause poor average response times. It is bad in situations where tasks have unequal lengths because short tasks may be held up by longer ones.

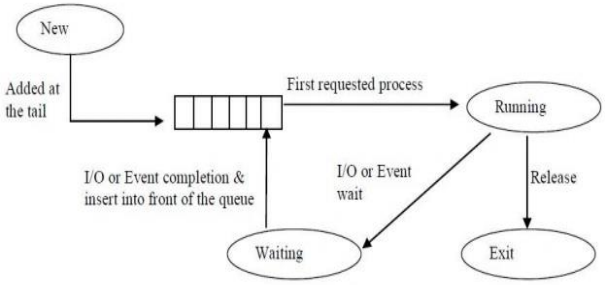

Fig. 2. State Diagram Showing the First-Come Come First-Served Scheduling.

Figure 2 illustrates that new processes are inserted at the rear of the ready queue, and they are strictly scheduled on a first-come first first-served basis. Once a process needs to do I/O or wait on an event, it moves into the Waiting state, and when it is done, it re-enters the queue at the front. Only the process at the front of the queue is allocated the Running state, and when the process, whether due to completion or release, enters the Exit state. Such a strict queue-based structure causes starvation of shorter or time-critical tasks in high-load situations.

2) Min-Min and Max-Min algorithms

Min-Min puts jobs that can be done quickly on the resources that can do them the fastest. This is done to get things done quickly. Max-Min schedules the longest tasks first to avoid delays and balance the overall workload:

- **Min-Min Algorithm:** The foundation of Min-Min scheduling is to run jobs on the resource that has the least minimal completion time (MCT) first, in order of priority. There are two stages to this method. The first step is to estimate how long it will take all of the resources to finish the meta job's tasks. [19]. The second step is to remove the job from the meta task and assign it to the resource with the least estimated completion time. This process is repeated until all of the tasks associated with the meta task have been mapped.
- **Max-Min Algorithm:** This approach corrects an issue with the min-min method. step two of the Max-Min algorithm is very different from the Min-Min method. In this step, instead of choosing a job based on its maximum completion time, the one with the lowest completion time is given to it. Both of the algorithm's initial stages are the same. This is why the Max-Min algorithm gets its name from it. Once the meta task is empty or all tasks have been mapped, the process is repeated.

3) Round Robin

The allocation of central processing unit (CPU) time among planned tasks served as the basis for the development of the Round Robin algorithm. Each job in the same environment receives a tiny fraction of a CPU core (Quantum, typically 10-100 milliseconds) and is added to a queue list [20]. One idea behind this approach is to use a ready queue to distribute CPU time across all the tasks that have been scheduled. The most important component of the Round Robin algorithm is the time slice (Time Quantum) that is allocated to each job that is submitted for execution. While many algorithms based on Round Robin propose using a static time quantum to distribute CPU time among all submitted jobs, this isn't always the best solution, even though it is an essential part of the process. Dynamic time quantum, which adjusts CPU time slices in response to changes in the ready queue for execution of tasks, is a more practical solution. In a same vein, this study proposes a method based on Round Robin that makes use of a dynamic time quantum and augments the current algorithm with a wiser layer to adapt CPU time to various scenarios.

IV. Optimization Techniques in Resource Allocation

Resource allocation, the process of making the best use of various and dynamic computer resources including memory, CPU, and bandwidth, is a big challenge in cloud computing. To address this, various optimization techniques are employed to enhance resource distribution, aiming to minimize make span, maximize throughput, and meet Quality of Service requirements. Heuristic methods are used for making quick decisions based on rules; metaheuristic algorithms are used to explore a broader solution space; and models based on artificial intelligence, like ML and reinforcement learning, learn optimal allocation strategies adaptively. Together, these approaches ensure high performance, cost-efficiency, and balanced workload distribution across cloud infrastructures.

G. Heuristic-Based Methods

Heuristic-based approaches are convenient, rule-of-thumb approaches to take resource allocation decisions in a swift and effective manner when the problem of achieving the optimal solution is either too complicated or time-consuming [21]. These methods do not exhaust all alternatives in the quest to find a good solution; they use clever shortcuts that achieve good enough results within a reasonable amount of time. In cloud computing, heuristics are useful in performing the task allocation to the available resources, considering aspects such as task size, deadline, or system load. They are thus very practical in the dynamic and large-scale systems.

H. Metaheuristic Algorithms

The word "meta" comes from Greek, meaning "beyond" or "higher level," while "heuristic" means "to discover" or "to investigate." Together, **meta-heuristics** refer to high-level strategies designed to guide and improve basic heuristic methods [22]. These techniques aim to find near-optimal solutions to complex problems like resource allocation, especially when exact methods are too slow or computationally expensive. While heuristic methods alone may not guarantee the best or even feasible solutions, meta-heuristics enhance their performance through intelligent search strategies, learning, and adaptation. They help balance speed and solution quality, making them ideal for dynamic and large-scale cloud environments.

I. AI and Machine Learning Methods of Resource Allocation

A ML-based approach to allocating resources. These goals can take the shape of optimizing the system's weighted sum rate, minimizing delay, improving power transmission efficiency, satisfying users' quality of service or quality of experience, or balancing resource use [23]. The details and layout of the strategy for allocating resources are defined by each of these goals. ML function in vehicle network resource allocation is covered in this section.

1) Supervised Learning

The use of supervised learning to allocate resources in vehicular networks serves a dual purpose. A supervised learning approach begins with using the regression problem to forecast the dataset's optimal allocation of resources. The second point is that supervised learning can be used to adjust to the ever-changing system's environment. Finally, a supervised learning algorithm can enhance the reinforcement learning algorithm's allocation choice effectiveness.

2) Unsupervised Learning

Unsupervised learning's function in ROP is to facilitate a cooperative approach wherein a set of vehicles sharing comparable attributes can enhance the ROP process. An intermediary vehicle or cluster head (CH) can be chosen to assist with data distribution by a central controller, like BS/RSU. Reducing scheduling overhead and data collision likelihood is possible with the cooperative method. The scalability and stability of the vehicular network can also be enhanced by implementing the clustering technology.

3) Reinforcement Learning

The learning phase and the technique of constructing a system model are similar across numerous articles on using RL for vehicle resource allocation. All four of these parts state, agent, action, and reward make up RL. The optimal policy for the allocation of resources can be built by this system using either a centralized or decentralized learning approach. The system's agent selection is affected by this learning model. In order to optimize its long-term payoff, an agent looks at the state space and decides what to do.

4) Multi-Objective Optimization Techniques

Multi-objective optimization techniques are smart strategies used to solve problems involving more than one goal, which often conflict with each other, like minimizing cost while maximizing performance. Instead of finding a single “perfect” solution, these techniques aim to find a balance or trade-off between objectives [24]. They generate a set of optimal solutions, known as the Pareto front, from which decision-makers can choose based on their priorities. These methods are widely used in resource allocation, scheduling, and the optimization of real-time systems.

V. Literature Review

This literature Summary section examines diverse strategies for cloud resource allocation, highlighting algorithms focused on execution time, cost-deadline trade-offs, virtualization, reliability, auction models, and load balancing to enhance performance, efficiency, and scalability in cloud computing environments.

Kumara et al. (2020) Time-Aware Resource Execution Algorithm (TRETA). The algorithm considers the total execution time of programs when deciding when to run them. Previous state-of-the-art heuristics for Make span, Degree of Imbalance, and System Throughput: Min-Max, FCFS, and MCT are compared with the approach using real-world workload traces of Nasa Ames iPSC/860. Workloads that include a wider range of tasks are used to run the tests in Cloud Sim 5.0 Simulator. Outperforming competing algorithms by a substantial margin in Make span, Degree of imbalance, and System Throughput is the recommended technique. Numerous customizable variants of the cloud platform's computing capabilities are available. Therefore, many tasks with varying data and computing needs can be executed on the cloud [25].

Vahidi and Rahmati (2019) aim to give due credit to the ground-breaking Grasshopper Optimization method (GOA) and to emphasize the importance of this method for efficient use of cloud computing resources. Eight datasets were utilized in the MATLAB simulation of the suggested approach to achieve this. For a more accurate assessment of GOA's performance, it was also compared against GA and SEIRA algorithms. The outcomes validated the usefulness of the suggested GOA and demonstrated its exceptional capacity to address the Cloud computing resource allocation issue, leading to the improvement of the identified solutions and, ultimately, opening doors to a satisfactory solution to the allocation problem. [26].

Win, Yee and Htoon (2019) proposes using linear programming to account for key parameters in order to determine the resource-minimized allocation model and apply it to the cloud's limited allocation constraints. One possible answer is the ORAM, which is derived from the Hungarian algorithm's modified cost matrix. This model is then tested experimentally to see how well it performs. A comparison with the current Hungarian-based allocation model, HABBP, and the traditional binding policy reveals that the suggested model achieves better results in terms of optimum execution time [27].

Alahmadi et al. (2019) ensure continuous service by integrating virtual continuity (VC), metro fog nodes, and the central cloud into a single design. To solve the problem of energy-efficient resource allocation, they developed a MILP model to optimise the distribution of work across the design's resources while keeping power consumption to a minimum. they examine the effects of various assignment algorithms on the utilization of virtual circuit resources and, by extension, on total power consumption, as it pertains to service provisioning under varied application demands. This study found that processing demands had less of an effect on power consumption than traffic demands. Power savings of up to 54% are possible when cloud-based architectures with metro fog nodes and vehicle edge nodes are integrated [28].

Alsmady et al. (2019) The cloud workflow scheduling issue is supposedly solved by using a Memetic Algorithm (MA). Finding a better way to schedule scientific activities in the cloud while keeping costs and deadlines in mind is the main objective. To complement the Genetic Algorithm (GA) and enhance individual solutions during global search, the suggested algorithm employed a hill climbing local search algorithm. Tests demonstrate that the suggested MA outperforms GA and PSO algorithms while simultaneously reducing the workflow's make span. Computations of a vast scale can be efficiently handled by utilizing cloud computing, a distributed computing system. "Pay as you go" is a popular concept for cloud computing that allows users to pay for the service as they use it [29].

Deshmukh and Amdani (2018) Virtualization, a key component of cloud computing, is essential for optimizing resource use among numerous users. Resources in a virtualized setting need to be overcommitted in order to maximize efficiency. When more resources are assigned to a virtual machine that is physically existent on the host, this is called overcommitment. The idea is that most VMs will only consume a tiny fraction of their reserved resources at any one moment. Virtualization makes it feasible to use various resources in a shared manner and

to do so at different times. These days, there is no shortage of methods that may be employed to better optimize the operation of the cloud system. Changes and strategies for memory optimization and management are studied in this research [30].

Table II synthesizes diverse research approaches in cloud resource allocation, emphasizing optimization algorithms, virtualization, reliability, and auction models, while addressing performance challenges and proposing future enhancements for scalable cloud environment.

Table 2: Summary of a Study on Resource Allocation in Cloud Computing.

Reference	Study On	Approach	Key Findings	Challenges	Future Direction
Kumara et al. (2020)	Optimal scheduling based on execution time	Cloud Sim 5.0 was used to test and compare TRETA, which stands for Total Resource Execution Time Aware Algorithm, with Min-Max, MCT, FCFS, and Min-Min.	Enhanced System Throughput, Degree of Imbalance, and Make span by utilizing real-world NASA traces	Handling complex heterogeneous workloads	Extend to multi-cloud environments and real-time dynamic scheduling
Vahidi et.al. (2019)	Optimization of resource allocation using Grasshopper Optimization Algorithm (GOA)	Simulation using MATLAB with 8 datasets; performance comparison with GA and SEIRA algorithms	GOA outperformed other algorithms in solving resource allocation problems, optimizing discovered responses effectively	Cloud computing: scalability and real-time applications	Continuous deployment in multi-cloud or hybrid setups; optimization for workloads that change over time
Win, et.al. (2019)	Identification and optimization of resource-minimized allocation model in cloud	Linear Programming (LP) for constraint definition; modified Hungarian Algorithm (ORAM) for allocation	ORAM outperformed traditional and preexisting approaches established on Hungary in terms of execution time.	Limited scope of real-world constraints in LP formulation	Extension to multi-objective optimization including QoS, latency, and cost parameters
Alahmadi et al. (2019)	Optimization of energy consumption in fog systems with virtual circuits - architecture in the cloud	MILP model; performance evaluation under varying application and traffic demands	Energy savings of up to 54% are possible with the integration of metro fog and vehicle edge nodes; processing needs have less of an effect on power consumption than traffic demands.	Complex coordination among fog, VC, and central cloud nodes	Development of adaptive models for real-time service provisioning under mobility and heterogeneity
Alsmady et al. (2019)	Scientific workflow scheduling	Memetic Algorithm (MA) combining GA with hill climbing local search to optimize cost and deadline	Superior performance compared to GA and PSO with a shorter manufacture span	Multi-objective optimization complexity	Apply to more diverse scientific workflows and integrate other local search methods
Deshmukh et.al. (2018)	Virtualization and memory optimization	Overcommitment strategy in virtualized environments	Efficient resource usage by allocating more than physically present resources	Risk of overloading physical hosts	Develop smarter overcommitment prediction models using AI

VI. Conclusion and Future Work

The present article has covered all the bases when it comes to cloud computing scheduling and resource allocation methods. System performance, cost-effectiveness, and user happiness are all affected by how well resources are managed in cloud infrastructures, which are becoming more and more dynamic and heterogeneous. Traditional static and centralized scheduling methods, while simple and efficient in stable environments, are limited by their lack of adaptability. In contrast, dynamic, decentralized, and intelligent scheduling approaches show greater promise in handling real-time workloads. Optimization strategies, particularly heuristic, metaheuristic, and ML-based methods, play a pivotal role in enhancing allocation accuracy and system responsiveness. ML, especially DL and reinforcement learning, offers adaptive and data-driven solutions to complex allocation problems, contributing to smarter, scalable, and more autonomous cloud systems.

However, most existing solutions are highly dependent on specific data formats or simulated environments, limiting their generalizability. Additionally, real-time adaptability remains a challenge due to computational complexity and fluctuating resource demands.

Integrating AI-driven optimization with static and dynamic scheduling methods is an area that might use more attention in future studies. There is also potential in exploring federated and

edge cloud environments where decentralized learning models can further reduce latency and improve scalability. Enhanced DL architectures and federated reinforcement learning can be developed to manage resource allocation in real-time across distributed cloud nodes. In addition, models should be designed with energy efficiency and quality of service (QoS) in mind. These models should be able to adapt to changing workloads and user needs with ease, all while keeping operational expenses to a minimum. The convergence of artificial intelligence (AI), cloud computing (CC), and the internet of things (IoT) presents an opportunity for smart, context-aware resource management systems.

References

1. G. Wei, A. Vasilakos, Y. Zheng, and N. Xiong, "A game-theoretic method of fair resource allocation for cloud computing services," *J. Supercomput.*, vol. 54, no. 2, pp. 252–269, Nov. 2010, doi: 10.1007/s11227-009-0318-1.
2. A. Kushwaha, P. Pathak, and S. Gupta, "Review of optimize load balancing algorithms in cloud," *Int. J. Distrib. Cloud Comput.*, vol. 4, no. 2, pp. 1–9, 2016.
3. S. Parikh, N. M. Patel, and H. B. Prajapati, "Resource Management in Cloud Computing: Classification and Taxonomy," 2017, doi: 10.48550/arXiv.1703.00374.
4. Z. Chen, L. Yang, Y. Huang, X. Chen, X. Zheng, and C. Rong, "PSO-GA-Based Resource Allocation Strategy for Cloud-Based Software Services With Workload-Time Windows," *IEEE Access*, vol. 8, pp. 151500–151510, 2020, doi: 10.1109/ACCESS.2020.3017643.
5. S. Garg, "AI-Driven Innovations in Storage Quality Assurance and Manufacturing Optimization," *Int. J. Multidiscip. Res. Growth Eval.*, vol. 1, no. 1, pp. 143–147, 2020, doi: 10.54660/.IJMRGE.2020.1.1.143-147.
6. G. E. Gonçalves *et al.*, "Resource allocation in clouds: concepts, tools and research challenges," *Minicursos - XXIX Simpósio Bras. Redes Comput. e Sist. Distrib.*, pp. 197–240, 2011.
7. S. S. S. Neeli, "Serverless Databases: A Cost-Effective and Scalable Solution," *Int. J. Innov. Res. Eng. Multidiscip. Phys. Sci.*, vol. 7, no. 6, p. 7, 2019.
8. H. Shukur, S. R. M. Zeebaree, R. R. Zebari, and D. Q. Zeebaree, "Cloud Computing Virtualization of Resources Allocation for Distributed Systems," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 2, pp. 98–105, Jun. 2020, doi: 10.38094/jastt1331.
9. P. Pathak, A. Shrivastava, and S. Gupta, "A Survey on Various Security Issues in Delay Tolerant Networks," *J. Adv. Shell Program.*, vol. 2, no. 2, pp. 12–18, 2015.
10. H. AlJahdali, A. Albatli, P. Garraghan, P. Townend, L. Lau, and J. Xu, "Multi-tenancy in Cloud Computing," in *2014 IEEE 8th International Symposium on Service Oriented System Engineering*, IEEE, Apr. 2014, pp. 344–351. doi: 10.1109/SOSE.2014.50.
11. A. Barnawi, S. Sakr, W. Xiao, and A. Al-Barakati, "The views, measurements and challenges of elasticity in the cloud: A review," *Comput. Commun.*, vol. 154, pp. 111–117, Mar. 2020, doi: 10.1016/j.comcom.2020.02.010.
12. A. Hameed *et al.*, "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems," *Computing*, vol. 98, no. 7, pp. 751–774, 2016, doi: 10.1007/s00607-014-0407-8.
13. N. Bulchandani, U. Chourasia, S. Agrawal, P. Dixit, and A. Pandey, "A survey on task scheduling algorithms in cloud computing," *Int. J. Sci. Technol. Res.*, vol. 9, no. 1, pp. 460–464, 2020.
14. T. A. Henzinger, A. V. Singh, D. Zufferey, V. Singh, and T. Wies, "Static scheduling in clouds," *3rd USENIX Work. Hot Top. Cloud Comput. HotCloud 2011*, 2011.
15. M. Alam, A. Khan, and A. K. Varshney, *A review of dynamic scheduling algorithms for homogeneous and heterogeneous systems*, vol. 732, no. May 2018. Springer Singapore, 2018. doi: 10.1007/978-981-10-8533-8_8.
16. D. Kaur and T. Sharma, "Scheduling Algorithms in Cloud Computing," *Int. J. Comput. Appl.*, vol. 178, no. 9, pp. 16–21, 2019, doi: 10.5120/ijca2019918801.
17. Y. Huang, N. Bessis, P. Norrington, P. Kuonen, and B. Hirsbrunner, "Exploring decentralized dynamic scheduling for grids and clouds using the community-aware scheduling algorithm," *Futur. Gener. Comput. Syst.*, vol. 29, no. 1, pp. 402–415, Jan. 2013, doi: 10.1016/j.future.2011.05.006.
18. N. Sharma, S. Tyagi, and S. Atri, "A Comparative Analysis of Min-Min and Max-Min Algorithms based on the Makespan Parameter," *Int. J. Adv. Res. Comput. Sci. ISSN No. 0976-5697*, vol. 8, no. 3, pp. 1038–1041, 2017, doi: 10.26483/ijarcs.v8i3.3151.
19. H. G. Tani and C. El Amrani, "Smarter Round Robin Scheduling Algorithm for Cloud Computing and Big Data," *J. Data Min. Digit. Humanit.*, pp. 1–8, Jan. 2018, doi: 10.46298/jdmdh.3104.
20. M. Tounsi, "A heuristic-based technique for university resource allocation problems," *2006 IEEE GCC Conf. GCC 2006*, pp. 1–6, 2006, doi: 10.1109/IEEEGCC.2006.5686243.
21. H. Madni, A. L. M. Shafie, C. Yahaya, and S. M. Abdulhamid, "An Appraisal of Meta-Heuristic Resource Allocation Techniques for IaaS Cloud," *Indian J. Sci. Technol.*, vol. 9, no. 4, Jan. 2016, doi: 10.17485/ijst/2016/v9i4/80561.
22. K. Zia, N. Javed, M. N. Sial, S. Ahmed, H. Iram, and A. A. Pirzada, "A Survey of Conventional and Artificial Intelligence / Learning based Resource Allocation and Interference Mitigation Schemes in D2D Enabled Networks," 2018.
23. S. Midya, A. Roy, K. Majumder, and S. Phadikar, "Multi-objective optimization technique for resource allocation and task scheduling in vehicular cloud architecture: A hybrid adaptive nature inspired approach," *J. Netw. Comput. Appl.*, vol. 103, pp. 58–84, 2018, doi: https://doi.org/10.1016/j.jnca.2017.11.016.
24. K. M. S. U. Bandaranayake, K. P. N. Jayasena, and B. T. G. S. Kumara, "An Efficient Task Scheduling Algorithm using Total Resource Execution Time Aware Algorithm in Cloud Computing," *2020 IEEE Int. Conf. Smart Cloud*, pp. 29–34, Nov. 2020, doi: 10.1109/SmartCloud49737.2020.00015.
25. J. Vahidi and M. Rahmati, "Optimization of Resource Allocation in Cloud Computing by Grasshopper Optimization Algorithm," in *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, IEEE, Feb. 2019, pp. 839–844. doi: 10.1109/KBEI.2019.8735098.
26. T. R. Win, T. T. Yee, and E. C. Htoon, "Optimized Resource Allocation Model in Cloud Computing System," in *2019 International Conference on Advanced Information Technologies (ICAIT)*, IEEE, Nov. 2019, pp. 49–54. doi: 10.1109/AITC.2019.8920852.

27. A. A. Alahmadi, M. O. I. Musa, T. El-Gorashi, and J. Elmirghani, "Energy Efficient Resource Allocation in Vehicular Cloud Based Architecture," in *2019 21st International Conference on Transparent Optical Networks (ICTON)*, IEEE, Jul. 2019, pp. 1–6. doi: 10.1109/ICTON.2019.8840547.
28. A. Alsmady, T. Al-Khraishi, W. Mardini, H. Alazzam, and Y. Khamayseh, "Workflow Scheduling in Cloud Computing Using Memetic Algorithm," in *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, 2019, pp. 302–306. doi: 10.1109/JEEIT.2019.8717430.
29. P. P. Deshmukh and S. Y. Amdani, "Virtual Memory Optimization Techniques in Cloud Computing," in *2018 International Conference on Research in Intelligent and Computing in Engineering (RICE)*, 2018, pp. 1–4. doi: 10.1109/RICE.2018.8509067.
30. *Chundru, S. K., Vangala, S. R., Polam, R. M., Kamarthapu, B., Kakani, A. B., & Nandiraju, S. K. K. (2024). A Machine Learning-Based Framework for Predicting and Improving Student Outcomes Using Big Educational Data (Approved by ICITET 2024 Conference Proceedings). Available at SSRN 5315635*.
31. *Nandiraju, S. K. K., Chundru, S. K., Vangala, S. R., Polam, R. M., Kamarthapu, B., & Kakani, A. B. (2025). Towards Early Forecast of Diabetes Mellitus via Machine Learning Systems in Healthcare. European Journal of Technology, 9(1), 35-50.*
32. Krutthika H. K. & Rajashekhara R. (2019). Network-on-chip: A survey on router design and algorithms. *International Journal of Recent Technology and Engineering*, *7*(6), 1687–1691. https://doi.org/10.35940/ijrte.F2131.037619
33. *Chalasani, R., Gangineni, V. N., Pabbineedi, S., Penmetsa, M., Bhumireddy, J. R., & Tyagadurgam, M. S. V. (2025). Big Data-Driven Approach for Lung Cancer Identification via Advanced Deep Transfer Learning Models. European Journal of Technology, 9(1), 51-67.*
34. *Vattikonda, N., Gupta, A. K., Polu, A. R., Narra, B., Buddula, D. V. K. R., & Patchipulusu, H. H. S. (2024). Machine Learning-Based Approaches for Detecting and Mitigating Distributed Denial of Service (DDoS) Attacks to Improved Cloud Security. European Journal of Technology, 8(6), 28-48.*
35. Krutthika H. K. & A.R. Aswatha. (2020). FPGA-based design and architecture of network-on-chip router for efficient data propagation. *IIOAB Journal*, *11*(S2), 7–25.
36. *Polu, A. R., Narra, B., Buddula, D. V. K. R., Hara, H., Patchipulusu, S., Vattikonda, N., & Gupta, A. K. Analyzing The Role of Analytics in Insurance Risk Management: A Systematic Review of Process Improvement and Business Agility.*
37. *Madhura, R., Varshitha, P., Nikitha, S., Niveditha, K. M., & Bhat, M. (2024, December). RTL design of 16-bit RISC Processor Using Vedic Mathematics. In 2024 IEEE 33rd Asian Test Symposium (ATS) (pp. 1-4). IEEE.*
38. Krutthika H. K. & A.R. Aswatha (2020). Design of efficient FSM-based 3D network-on-chip architecture. *International Journal of Engineering Trends and Technology*, *68*(10), 67–73. https://doi.org/10.14445/22315381/IJETT-V68I10P212
39. *Harinandan, R., Kumar, M., Vamshi, P., Padma, C. R., Krishnappa, K. H., & Raghunandan, J. R. (2024, August). Design and Development of a Real-time Monitoring System for ACL Injury Prevention. In 2024 2nd International Conference on Networking, Embedded and Wireless Systems (ICNEWS) (pp. 1-6). IEEE.*
40. *Krishnappa, K. H. (2024). Traffic pattern analysis for malicious node detection in NoC design. Journal of Communications, 9, 12.*
41. *Mukund Sai Vikram Tyagadurgam, Venkataswamy Naidu Gangineni, Sriram Pabbineedi, Mitra Penmetsa, Jayakeshav Reddy Bhumireddy, et al. (2024) AI-Powered Cybersecurity Risk Scoring for Financial Institutions Using Machine Learning Techniques. Journal of Artificial Intelligence & Cloud Computing. SRC/JAICC-482. DOI: doi.org/10.47363/JAICC/2024(3)452.*
42. HK, K. (2020). Design of Efficient FSM Based 3D Network on Chip Architecture. *INTERNATIONAL JOURNAL OF ENGINEERING*, *68*(10), 67-73.
43. Krutthika, H. K. (2019, October). Modeling of Data Delivery Modes of Next Generation SOC-NOC Router. In *2019 Global Conference for Advancement in Technology (GCAT)* (pp. 1-6). IEEE.
44. Ajay, S., Satya Sai Krishna Mohan G, Rao, S. S., Shaunak, S. B., Krutthika, H. K., Ananda, Y. R., & Jose, J. (2018). Source Hotspot Management in a Mesh Network on Chip. In *VDAT* (pp. 619-630).
45. Nair, T. R., & Krutthika, H. K. (2010). An Architectural Approach for Decoding and Distributing Functions in FPUs in a Functional Processor System. *arXiv preprint arXiv:1001.3781*.
46. Gopalakrishnan Nair, T. R., & Krutthika, H. K. (2010). An Architectural Approach for Decoding and Distributing Functions in FPUs in a Functional Processor System. *arXiv e-prints*, arXiv-1001.
47. Krutthika H. K. & A.R. Aswatha. (2021). Implementation and analysis of congestion prevention and fault tolerance in network on chip. *Journal of Tianjin University Science and Technology*, *54*(11), 213–231. https://doi.org/10.5281/zenodo.5746712.
48. *Pabbineedi, S., Kakani, A. B., Nandiraju, S. K. K., Chundru, S. K., Tyagadurgam, M. S. V., & Gangineni, V. N. (2023). Scalable Deep Learning Algorithms with Big Data for Predictive Maintenance in Industrial IoT. International Journal of AI, BigData, Computational and Management Studies, 4(1), 88-97.*
49. *Chalasani, R., Vangala, S. R., Polam, R. M., Kamarthapu, B., Penmetsa, M., & Bhumireddy, J. R. (2023). Detecting Network Intrusions Using Big Data-Driven Artificial Intelligence Techniques in Cybersecurity. International Journal of AI, BigData, Computational and Management Studies, 4(3), 50-60.*
50. *Vangala, S. R., Polam, R. M., Kamarthapu, B., Penmetsa, M., Bhumireddy, J. R., & Chalasani, R. (2023). A Review of Machine Learning Techniques for Financial Stress Testing: Emerging Trends, Tools, and Challenges. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 4(1), 40-50.*
51. *Kakani, A. B., Nandiraju, S. K. K., Chundru, S. K., Tyagadurgam, M. S. V., Gangineni, V. N., & Pabbineedi, S. (2023). A Survey on Regulatory Compliance and AI-Based Risk Management in Financial Services. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 4(4), 46-53.*

52. *Bhumireddy, J. R., Chalasani, R., Vangala, S. R., Kamarthapu, B., Polam, R. M., & Penmetsa, M. (2023). Predictive Machine Learning Models for Financial Fraud Detection Leveraging Big Data Analysis. International Journal of Emerging Trends in Computer Science and Information Technology, 4(1), 34-43.*
53. *Gangineni, V. N., Pabbineedi, S., Kakani, A. B., Nandiraju, S. K. K., Chundru, S. K., & Tyagadurgam, M. S. V. (2023). AI-Enabled Big Data Analytics for Climate Change Prediction and Environmental Monitoring. International Journal of Emerging Trends in Computer Science and Information Technology, 4(3), 71-79.*
54. *Polam, R. M. (2023). Predictive Machine Learning Strategies and Clinical Diagnosis for Prognosis in Healthcare: Insights from MIMIC-III Dataset. Available at SSRN 5495028.*
55. *Narra, B., Gupta, A., Polu, A. R., Vattikonda, N., Buddula, D. V. K. R., & Patchipulusu, H. (2023). Predictive Analytics in E-Commerce: Effective Business Analysis through Machine Learning. Available at SSRN 5315532.*
56. *Narra, B., Buddula, D. V. K. R., Patchipulusu, H. H. S., Polu, A. R., Vattikonda, N., & Gupta, A. K. (2023). Advanced Edge Computing Frameworks for Optimizing Data Processing and Latency in IoT Networks. JOETSR-Journal of Emerging Trends in Scientific Research, 1(1).*
57. *Patchipulusu, H. H. S., Vattikonda, N., Gupta, A. K., Polu, A. R., Narra, B., & Buddula, D. V. K. R. (2023). Opportunities and Limitations of Using Artificial Intelligence to Personalize E-Learning Platforms. International Journal of AI, BigData, Computational and Management Studies, 4(1), 128-136.*
58. *Madhura, R., Krishnappa, K. H., Shashidhar, R., Shwetha, G., Yashaswini, K. P., & Sandya, G. R. (2023, December). UVM Methodology for ARINC 429 Transceiver in Loop Back Mode. In 2023 3rd International Conference on Mobile Networks and Wireless Communications (ICMNWC) (pp. 1-7). IEEE.*
59. *Shashidhar, R., Kadakol, P., Sreeniketh, D., Patil, P., Krishnappa, K. H., & Madhura, R. (2023, November). EEG data analysis for stress detection using k-nearest neighbor. In 2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS) (pp. 1-7). IEEE.*
60. *KRISHNAPPA, K. H., & Trivedi, S. K. (2023). Efficient and Accurate Estimation of Pharmacokinetic Maps from DCE-MRI using Extended Tofts Model in Frequency Domain.*
61. *Krishnappa, K. H., Shashidhar, R., Shashank, M. P., & Roopa, M. (2023, November). Detecting Parkinson's disease with prediction: A novel SVM approach. In 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE) (pp. 1-7). IEEE.*
62. *Shashidhar, R., Balivada, D., Shalini, D. N., Krishnappa, K. H., & Roopa, M. (2023, November). Music Emotion Recognition using Convolutional Neural Networks for Regional Languages. In 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE) (pp. 1-7). IEEE.*
63. *Madhura, R., Krishnappa, K. H., Manasa, R., & Yashaswini, K. P. (2023, August). Slack Time Analysis for APB Timer Using Genus Synthesis Tool. In International Conference on ICT for Sustainable Development (pp. 207-217). Singapore: Springer Nature Singapore.*
64. *Krishnappa, K. H., & Gowda, N. V. N. (2023, August). Dictionary-Based PLS Approach to Pharmacokinetic Mapping in DCE-MRI Using Tofts Model. In International Conference on ICT for Sustainable Development (pp. 219-226). Singapore: Springer Nature Singapore.*
65. *Krishnappa, K. H., & Gowda, N. V. N. (2023, August). Dictionary-Based PLS Approach to Pharmacokinetic Mapping in DCE-MRI Using Tofts Model. In International Conference on ICT for Sustainable Development (pp. 219-226). Singapore: Springer Nature Singapore.*
66. *Madhura, R., Krutthika Hirebasur Krishnappa. et al., (2023). Slack time analysis for APB timer using Genus synthesis tool. 8th Edition ICT4SD International ICT Summit & Awards, Vol.3, 207–217. https://doi.org/10.1007/978-981-99-4932-8_20*
67. *Shashidhar, R., Aditya, V., Srihari, S., Subhash, M. H., & Krishnappa, K. H. (2023). Empowering investors: Insights from sentiment analysis, FFT, and regression in Indian stock markets. 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE), 01–06. https://doi.org/10.1109/AIKIIE60097.2023.10390502*
68. *Jayakeshav Reddy Bhumireddy, Rajiv Chalasani, Mukund Sai Vikram Tyagadurgam, Venkataswamy Naidu Gangineni, Sriram Pabbineedi, Mitra Penmetsa. Predictive models for early detection of chronic diseases in elderly populations: A machine learning perspective. Int J Comput Artif Intell 2023;4(1):71-79. DOI: 10.33545/27076571.2023.v4.i1a.169*