Enhancement of containerization-based cloud computing performance using hybrid metaheuristic optimization algorithm: A review

G.Ramasamy, Research Scholar
Department of Computer Science
School of Computing Sciences
Vels Institute of Science, Technology & Advanced Studies
Pallavaram, Chennai, India
govindan.ramasamy@gmail.com

Dr.S. Kamalakkannan, Professor
Department of Information Technology
School of Computing Sciences
Vels Institute of Science, Technology & Advanced Studies
Pallavaram, Chennai, India
kannan.scs@vistas.ac.in

***Abstract*—Container-based virtualisation has become a cornerstone of Cloud Computing (CC) due to its lightweight and scalable properties when compared to traditional virtual machines. However, it is still difficult to optimising resource allocation in containerised systems, and current approaches frequently fall short in addressing problems like high computational costs, scalability, cold-start scenarios, and Graphics Processing Unit (GPU) management. Container resource allocation is critical for effectively managing computing workloads in cloud settings, enabling scalability, performance optimization, and cost effectiveness. It is extensively used in a various industry, including e-commerce, healthcare, and finance, where dynamic workloads call for efficient resource management. Traditional resource allocation techniques, while their importance, frequently face difficulties such as unequal load distribution and resource underutilization. To address these constraints, this paper suggests a hybrid metaheuristic optimization algorithm that addresses limitations such as network distance, execution time, and migration costs to provide balanced load distribution. This study attempts to analyse the efficiency, time consumption and latency of resource allocation in dynamic cloud systems by addressing existing limitations.**

Keywords*— *Cloud computing, Virtualization, optimization, container resource allocation.

I. INTRODUCTION

CC facilitates the delivery of information services and network computing resources such as apps, servers, and storage via the internet without the need for installation or purchase. Intel, IBM, and several other American businesses, including academic institutions, started running a CC virtual laboratory in 2005. This kind of business started with a few trials at North Carolina State University, which is close to IBM's headquarters. Google and IBM partnered in 2007 to begin processing a novel network computing technique known as CC [1]. After testing, a significant number of research organisations became interested in the novel conversional Intel and Microsoft computing method. A computational approach known as CC offers virtualised resources and dynamic scalability as an online service. This is the combination of several technologies, including virtualisation, parallel computing, distributed computing, and utility informatics. It might be categorised as hybrid, private, and public. SAAS, PAAS, and IAAS are the three service levels seen in cloud-based computing environments. Nowadays, one of the most popular fundamental CC methods is virtualisation. CC may benefit from virtualisation technology. The advantages of reducing energy use, equipment expenditures, and operating management expenses are becoming more and more available to enterprises. Data centres for various industries, including government, finance, and education, have recently made substantial use of cloud media virtualization-based technologies. Numerous benefits of CC include democratising access to easier maintenance, lower IT expenses, scalability, security, dependability, and customisation. CC services are shown in Figure 1.

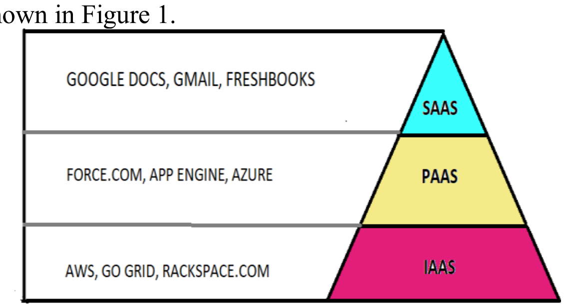

FIGURE 1 CLOUD COMPUTING SERVICES

VMware is a method of creating a digital "version" or representation of a server, system software, storage devices, or networking components that may be utilised on numerous nodes simultaneously [2]. It appears that the primary goal of virtualisation is to control processes by transforming conventional computers into ones that are far more scalable, efficient, and high-end [3]. Virtualised operating systems, cloud servers, and hardware CC are examples of automation. Virtualisation is the process by which anything is built virtually rather than in real time. A technology that reduces costs and resources, virtualisation is continuously changing the fundamentals of computing. Virtual Machines (VMs) are the result of combining hardware and software innovation, allowing many operating systems to run on one system. CC stands out as the key advancement driving information technology today. The creation of operating systems, virtual hardware, software, platforms, storage, and network devices as opposed to real hardware is known as virtualisation in computing. As the virtual environment changes more quickly than the real one, IT companies that operate in virtualised environments must adjust to and manage a lot of changes. Virtualisation makes CC accessible and flexible [4].

Servers are consolidated, resources are allocated, costs are minimized, and scalability along with flexibility is enhanced through virtualization [5]. It also supports disaster recovery, ensures continuity in business operations, aids in testing and development, facilitates remote work, strengthens application management, advances IT training, and streamlines cloud migration and usage [6, 7].

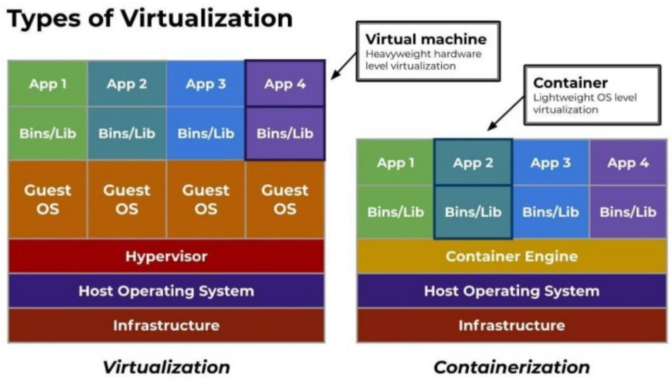

FIGURE 2 VIRTUALIZATION MODELS

In CC, there are primarily two kinds of virtualisation approaches. One is software virtualisation, and the other is hardware virtualisation. Software is used to construct an abstraction layer over the actual hardware in hardware virtualisation. In the process, a VM is created, which is a virtual replica of a computing system. It is now possible for a single physical system to operate many virtual machines. An actual physical machine can be represented virtually as a virtual machine. The resources available determine how many VMs can be built from a single physical computer. Operating systems may differ among the VMs on a same system. The virtual machine is created by a software layer known as a hypervisor. Each VM is assigned the necessary hardware, including memory, storage, and processing power, and any interaction between them is blocked by a hypervisor. A hypervisor is used by all VMs to communicate with the host computer. Containerisation, often known as software virtualisation, is the second virtualisation strategy. Figure 2 shows the virtualization models.

This study concentrated on improving the containerization-based CC performance by managing hypervisor capability in identifying the requirement of hardware using precise VMs optimization through combination of metaheuristic optimization algorithms to obtain better performance. Moreover, session 2 addresses the several optimizations in virtualization of resource allocation in virtualizing cloud infrastructure. Session 3 discusses the research methodology of best individual metaheuristic optimization method such as Hippopotamus Algorithm and Pufferfish Optimization Algorithm. Session 4 discusses the result and discussion for hybrid metaheuristic optimization as well as individual metaheuristic optimization algorithm for performance analysis. Session 5 discusses the conclusion about hybrid metaheuristic optimization algorithm for better performance in virtualizing cloud infrastructure.

II. LITERATURE REVIEW

Virendra et al., have examined the virtualisation technology and its significance in CC, and the work provides a thorough analysis of the function of virtualisation. An essential component of CC is virtualisation, which makes it possible to abstract and share physical resources among separate VMs. However, enterprise cloud adoption is still hampered by security concerns, particularly as virtualisation creates additional attack surfaces. The host operating system, virtual networks, administrative level and hypervisor are all susceptible to compromise, even though VMs offer robust workload isolation. Secure isolation is essential in multi-tenant cloud environments. Recent studies on protecting virtualised cloud infrastructure and reducing multi-tenancy hazards are examined in this review of the literature [8] [9].

Goel et al., have investigated the challenges associated with the adoption of cloud virtualisation, including network obstacles, reliability issues, efficiency expenses, and the complexity of security operations. It is still necessary to make more advancements in cloud-native software design, supplementary technologies like containers, and hypervisors. It's still difficult to realise virtualization's full potential [10]. Huang et al. offered a hybrid approach that uses partial homomorphic security and data anonymisation to facilitate cloud-based mining of sensitive datasets with maintaining privacy. Yet, scaling to massive data workloads effectively is still a challenge. Maintaining privacy is essential for cloud analytics [11].

Liu., have assessed cloud-based network virtualisation software optimisations. Throughput is accelerated by improvements like multi-queue optimisation and vector packet processing. Performance is also customised by automated configuration adjustment. Nonetheless, portable designs are necessary due to hardware requirements. Scalability and performance of the cloud are enhanced by effective virtualisation [12]. Morabito., have examined container technology for IoT gateways. Containers offer quick startup speeds and lightweight virtualisation, making them ideal for applications on the edge. However, optimised configurations are necessary due to limited resources. Cloud capabilities are extended to edge devices through virtualisation [13].

Parast et al., have investigated threats in various service models, including IaaS, PaaS, and SaaS. Common problems include privilege abuses, shared environments, and network exposures. It is advised to use a multi-layered strategy that incorporates monitoring, encryption, and access controls. Defences are informed by knowledge of cloud security threats [14].

According to Zhou et al., traditional resource allocation methods in CC primarily use static or rule-based approaches. These methods employ predefined policies and heuristics to distribute resources among tasks and applications. The round-robin algorithm ensures fairness by allocating resources in a circular order but often lacks efficiency. Greedy algorithms optimize allocation by making locally optimal choices at each step. While effective in specific scenarios, these methods struggle to adapt to dynamic cloud workloads and heterogeneous resources [15].

Singh et al., have stated that adapting resource sharing through optimisation techniques is another crucial aspect of related research. Techniques like reinforcement learning, ant colony optimisation, and genetic algorithms are used to adjust resource usage in response to shifting task trends. In addition to making resource allocation decisions more effective, these techniques also increase system scalability and resilience across a range of cloud environments. The integration of AI with other emerging technologies, such edge computing and Internet of Things (IoT) devices has also being studied by researchers. By utilising AI to distribute resources at the edge, cloud providers can decrease energy and data consumption while simultaneously enhancing the performance of latency-sensitive applications [16]. New techniques, like as game theory and economic models, have also been developed as a result of research into how to manage cloud resources effectively and economically. In order to determine the most effective ways to distribute resources and improve market performance, these systems examine pricing models, cost considerations, and the manner in which cloud service providers compete with one another.

Zhong and Buyya have proposed a new Cloud Cost Optimization strategy for cost efficient Kubernetes container orchestration using Heterogenous Task Allocation using Elastic Computing Resources contains three key characteristics, the first is integrating the containers with the resources that are already available using task packing. Second is using scaling algorithms, adjust the cluster size for varying workloads and third one involves in establishing a rescheduling strategy to find and turn off unused virtual machine instances with the goal of cost savings, while seamlessly reallocating pertinent jobs without compromising task progress. When this strategy compared to default Kubernetes cost, it results in lowering costs overall by 23% to 32% on different workloads and patterns [17].

III. PROBLEM STATEMENT

Traditional container resource allocation methods in CC face significant limitations that hinder their efficiency and scalability. These methods often fail to account for the dynamic and heterogeneous nature of cloud environments, resulting in suboptimal resource utilization. For instance, many conventional algorithms lack the ability to manage GPU resources effectively, which is increasingly essential for microservice applications requiring high-performance computation. Furthermore, approaches reliant on historical data for resource prediction, such as FoRES, are unsuitable for cold start scenarios where such data is unavailable, limiting their applicability. Other models neglect critical factors such as computational time and fail to adapt dynamically to changes in the cloud environment, such as node movements. Additionally, traditional clustering-based techniques, while useful for container categorization, struggle with scalability and computational efficiency, particularly when addressing large-scale datasets. These limitations highlight the need for advanced approaches that combine dynamic adaptability, efficient resource prediction, and scalability to improve container resource allocation in modern CC environments. Table 1 shows the various optimization algorithms and its results.

TABLE 1. COMPARATIVE EVALUATION OF OPTIMIZATION ALGORITHMS

Author	Algorithm	Findings	Limitations
Ibrahim et al. [18]	Power-aware particle swam optimization approach	Reduction of 8.01% in power consumption, 39.65% in migration rates, 66.33% in host deactivations, and 11.87% in Service Level Agreement (SLA) breaches.	Testing the proposed strategy in real-world CC settings is essential. Optimizing performance requires careful consideration of factors such as storage, network bandwidth, and processing capacity.
Barthwal et al. [19]	Ant colony optimization	Compared to the power-aware best-fit approach, the proposed technique consumed minimal energy while ensuring adherence to SLA requirements.	Performance can be improved further by using additional resources such as memory, storage and bandwidth
Gharehpasha et al. [20]	Whale optimization algorithm	The suggested technique improves resource utilization while using less energy in the cloud data centre	To fully utilize virtualization's advantages, security and privacy issues must be addressed. Cloud task scheduling and the use of multiple cloud platforms should be taken into account.

Gomathi et al. [21]	Pareto-based composite mutation particle swarm optimization.	The proposed method addresses the SLA violations and limited energy usage	The proposed method must be accessed using a real-time operating CC environment.
Kanagaraj et al. [22]	Elephant herding optimization for identifying host overload and underload conditions.	The suggested approach resulted in a 0.073% decrease in VM migrations, an 11% reduction in energy consumption, and a 6.15% decline in SLA violations.	More assessments based on actual workload traces should be conducted, and factors such as network bandwidth or storage should also be considered.
Wang et al. [23]	Improved Hippopotamus Algorithm	Enhances prediction accuracy while also exhibiting robust generalization abilities.	The research excludes updated optimization models along with decomposition denoising techniques for improving prediction accuracy.
Malvi and Sateesh [24]	Hippopotamus Optimization Algorithm	It optimizes hidden neurons, boosting accuracy to an impressive 0.9831.	lack of consideration for channel sparsity, which could potentially reduce the complexity of MIMO channel estimation
Al-Baik et al. [25]	Pufferfish Optimization Algorithm	POA offers parameter-free design, high efficiency in diverse problems, balanced search for rapid convergence, and robust performance in real-world applications.	it cannot guarantee achieving the global optimum, and its effectiveness is uncertain, with the potential for better-performing algorithms to emerge.

IV. PROPOSED METHODOLOGY

Research Methodology

This study proposes a hybrid optimization approach that combines the Hippopotamus Optimization Algorithm (HOA) and the Pufferfish Optimization Algorithm (POA) to improve resource allocation in cloud computing. The methodology is divided into three distinct phases: system modelling, resource demand prediction, and hybrid optimization. Phase 1: System Modeling The first phase focuses on developing a comprehensive architecture for containerized resources in cloud computing. This involves detailing the resource allocation mechanisms and capturing the dynamic and heterogeneous nature of cloud environments. By modeling the system, we create the foundation for effective resource management that can adapt to various cloud environments. Phase 2: Resource Demand Prediction: In the second phase, we employ advanced machine learning techniques to predict the future resource requirements of containers. Using both historical and real-time data, the system forecasts resource demands, allowing for accurate predictions that support proactive scaling and optimal resource management. This predictive capability is crucial for ensuring that resources are allocated efficiently before any shortage or surplus occurs. Phase 3: Hybrid Optimization Algorithm: The final phase integrates the HOA and POA algorithms to optimize resource allocation. The HOA algorithm first explores various allocation possibilities to ensure workloads are evenly distributed, avoiding both server overloading and underutilization. After identifying potential solutions, the POA algorithm fine-tunes these allocations, adjusting them to minimize delays and improve overall system efficiency. Inspired by the pufferfish's defense mechanism, the POA algorithm quickly adapts to changes in workload, enhancing the system's responsiveness to real-time demands.

This hybrid optimization process aims to select the best resource allocation strategy based on key factors such as execution speed, efficiency, and resource demand. It continuously learns from ongoing operations and adapts for improved accuracy and performance. This research addresses the limitations of traditional container resource allocation methods in cloud computing by proposing an integrated, three-phase methodology. By combining system modeling, demand prediction, and hybrid optimization, the approach ensures balanced resource utilization, minimizes computational overhead, and adapts to fluctuations in workload. This comprehensive solution offers a more efficient and scalable approach to cloud resource management, particularly in the face of dynamic workloads and varying network conditions.

Hippopotamus Optimization Technique

The HOT for container resource allocation in cloud architecture is inspired by the energy-efficient and cooperative behaviors of hippopotamuses. The methodology begins by modelling the resource allocation problem as an optimization task with the objective of minimizing resource wastage, ensuring balanced utilization, and meeting Quality of Service (QoS) requirements. The available resources in cloud nodes and the container demands are defined, and the optimization goal is to allocate requested resource to Cloud Computing Containers (CCC) while adhering to constraints such as resource capacity and QoS thresholds. The process initiates by creating a population of candidate solutions, with each solution corresponding to a possible strategy for resource allocation. Each solution is encoded as a vector, and its fitness is evaluated based on the optimization objective, which combines cost efficiency and QoS penalties.

HOT operates in two phases: exploration and exploitation. During the exploration phase, the algorithm simulates the energy-efficient and movement behaviors of hippopotamuses by introducing randomization to diversify candidate solutions and adjusting them toward promising regions in the search space. This prevents premature convergence and ensures thorough exploration of potential solutions. In the exploitation phase, the algorithm mimics the cooperative nature of hippopotamuses by refining the best solutions through local search and adjustment mechanisms. Solutions share information to guide the optimization process, ensuring that resource allocation is optimized for efficiency and performance. The fitness function evaluates the trade-off between minimizing operational costs and satisfying QoS requirements.

Exploration Phase:

In this phase, inspired by the energy-efficient foraging movements of hippopotamuses, the algorithm explores the search space for diverse resource allocation possibilities. Candidate solutions are adjusted using a combination of random and directional updates, simulating hippopotamus behavior in moving toward favourable resource regions. The movement of a solution X_i is updated in equation 1:

$$X_i^{t+1} = X_i^t + \alpha(G - X_i^t) + \beta . rand\ () \quad (1)$$

where X_i^{t+1} new position of the i^{th} hippopotamus at iteration t+1, X_i^t is the current solution, G is the global best solution, α and β are control parameters influencing the movement, and rand () introduces randomness.

V. Exploitation Phase

Once promising regions in the search space are identified, HOT focuses on fine-tuning solutions to converge toward an optimal resource allocation. Inspired by the cooperative behavior of hippopotamuses, the solutions exchange information to improve the overall fitness. This phase balances local search and exploitation by refining resource allocation configurations and ensuring better utilization of available resources.

The fitness of each candidate solution is computed using a fitness function that evaluates resource efficiency and QoS satisfaction. The fitness function *F(X)* is defined in equation 2:

$$F(X)=w_1 . Cost(X) + w_2 . QoSPenalty(X) \quad (2)$$

Where, F(X) is Fitness Function Value, Cost(X) is Resource Cost Function, QoS Penalty(X) is QoS Penalty Function, $w_1 and w_2$ are weighting factors for cost and QoS penalties respectively.

Pufferfish optimization technique

The Tetraodontidae and Tetraodontiformes families of fishes include pufferfish, which are mostly marine and estuary in nature. This fish shares morphological similarities with large-spiked porcupinefish. Pufferfish have tiny to medium bodies, and they can reach up to 50 centimetres in length. One of pufferfish's most distinguishing characteristics is its beak-like four teeth. Additionally, pufferfish are peculiar in that they lack ribs, a pelvis, and pectoral fins. The specialised defence mechanism of the pufferfish, which involves sucking water via the mouth cavity, is the reason for the considerable loss of fin and bone structures.

The pufferfish are a prime prey for predators as they move extremely slow. The pufferfish uses its unique defence mechanism to swell its elastic stomach with water until it forms a big, spherical, spiky ball that can fend off predator attacks. Instead of presenting an easy meal, pufferfish's pointed spines present a ball of sharp points to the eager predator. This warning alerts predators to the danger and makes them avoid the pufferfish. Conflicts with predators and the usage of the defence mechanism of transforming into an object made of pointed spines in response to predator attacks are two of the pufferfish's most important natural behaviours. The design of the suggested POA technique, which is covered below, makes use of models of these natural processes, which include

(i) A predator attack on a pufferfish
(ii) Pufferfish defence mechanism against predator attacks.

The predator attack and defense mechanisms in the Pufferfish Optimization Algorithm (PFOA) are inspired by the real-world behavioural strategies of pufferfish to ensure survival. These mechanisms are abstracted into the optimization process to balance exploration (searching broadly in the solution space) and exploitation (fine-tuning solutions in promising areas). Below is a description of how these mechanisms are modelled.

1. Predator Attack Mechanism (Exploration Phase)

In the wild, predators pose a threat to pufferfish, prompting them to adopt defensive strategies. In PFOA, this translates into the algorithm's ability to explore new regions of the search space. When predators (or challenges) are detected, the pufferfish "inflates" to evade or confuse them.

Mathematically, this is modelled as a global search for potential solutions by expanding the population dynamically:

$$x_i' = x_i + r_1 \,.\, (u - l), \tag{3}$$

where:

- x_i' is the new solution generated by exploration,
- x_i is current position of the i^{th} Pufferfish,
- r_1 is a random number in [0,1],
- u and l represent the upper and lower bounds of the solution space.

This mechanism encourages the algorithm to explore unvisited regions, avoiding premature convergence.

2. Pufferfish Defence Mechanism (Exploitation Phase)

When under attack, pufferfish contract after inflating, making themselves less visible or maneuverable to predators. This behaviour is adapted to intensify the search around high-quality solutions by refining them iteratively. The defense mechanism prioritizes exploiting promising regions to achieve better solutions.

Mathematically, this is expressed as:

$$x_i' = x_i + r_2 \,.\, (x_{\text{best}} - x_i), \tag{4}$$

where:

- x_i' is the refined solution,
- x_i is current position of the i^{th} Pufferfish,
- x_{best} is the best solution found so far,
- r_2 is a contraction coefficient in [0,1].

This intensification mechanism ensures the algorithm converges effectively toward optimal solutions by focusing on areas with higher fitness.

The HOA and POA can be further improved by:

- Adaptive Learning Techniques: Integrating reinforcement learning-based parameter tuning to dynamically adjust algorithm parameters according to workload variations, reducing computational overhead.
- Parallel Processing in Optimization: Implementing parallel execution of optimization algorithms to accelerate convergence speed and reduce time consumption.
- Network-Aware Container Placement: Optimizing resource allocation by considering data transmission delays and network distance, ensuring reduced latency and improved performance.
- Preemptive Resource Allocation: Introducing predictive models that anticipate workload demand, thereby minimizing cold-start issues and enhancing scalability.

VI. RESULT AND DISCUSSION

Figure 3 illustrates the comparison of time consumption of five optimization algorithms, showing that some take considerably more time than others. The Artificial Bee Colony and Grasshopper algorithm [26] has the shortest time consumption in seconds.In container resource allocation for cloud computing, lower time consumption indicates faster decision-making and efficient resource utilization, crucial for dynamic and scalable environments. Higher time consumption can delay resource provisioning, affecting system performance and responsiveness.

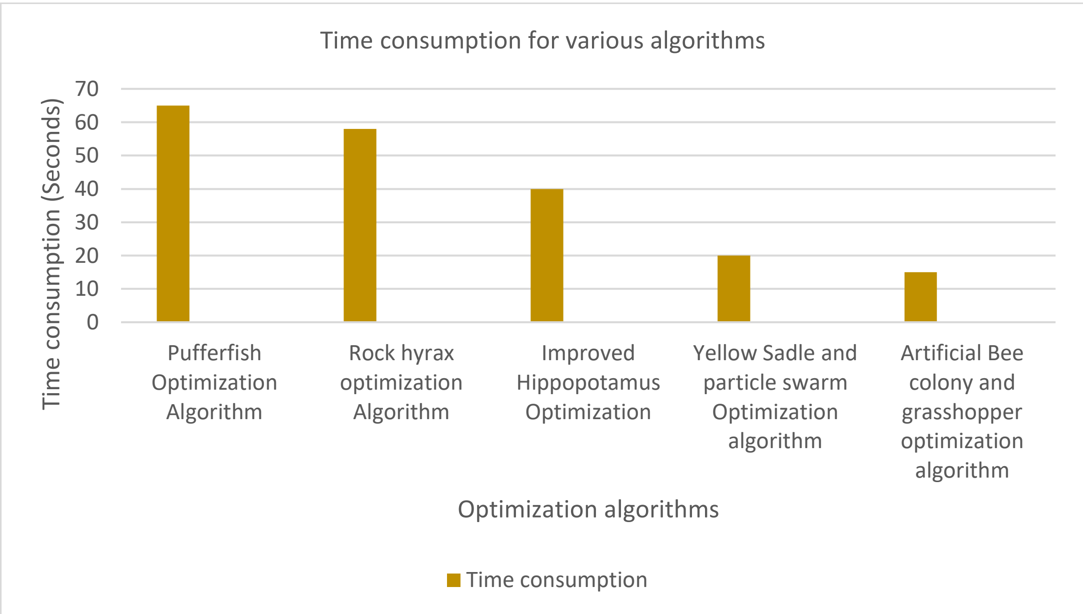

FIGURE 3 TIME CONSUMPTION OF VARIOUS OPTIMIZATION ALGORITHMS

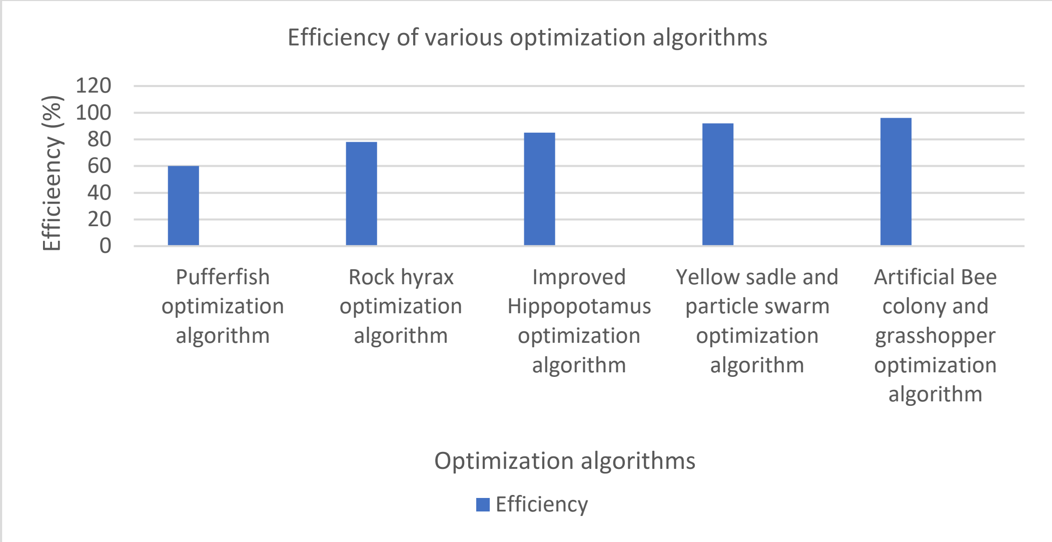

FIGURE 4 EFFICIENCY FOR VARIOUS OPTIMIZATION ALGORITHMS

Figure 4 illustrates the efficiency of five optimization algorithms for container resource allocation in terms of energy usage. The Artificial Bee Colony and Grasshopper algorithm [26] achieves the highest energy efficiency (%), followed by the Yellow Saddle and particle swarm optimization [27]. This means these algorithms manage resources effectively while minimizing energy consumption.

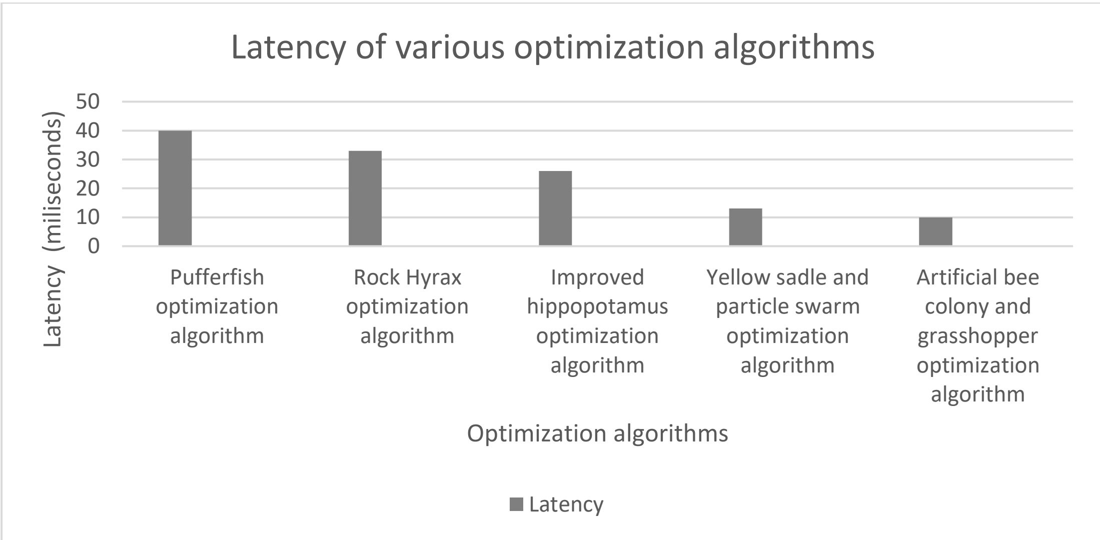

FIGURE 5 LATENCY OF VARIOUS OPTIMIZATION ALGORITHMS

Figure 5 compares the latency, which is the time delay in processing or computing a solution, for various optimization algorithms. The Pufferfish Optimization Algorithm [25] exhibits the highest latency, followed by the Rock Hyrax [28] and Improved Hippopotamus Optimization [23] Algorithms. The Yellow Saddle and particle swarm optimization algorithm [27] shows moderate latency, while the Artificial Bee Colony and Grasshopper Optimization Algorithm [26] has the lowest latency, indicating it is the fastest and most efficient among the compared methods. The proposed hybrid optimization method improves accuracy by predicting resource demands, balancing workloads, and reducing delays in cloud computing. The HOA helps explore different ways to distribute resources, while the POA fine-tunes the allocation process for better efficiency. By using real-time and past data, the system ensures optimal resource usage with minimal waste. The results are validated by comparing the proposed hybrid optimization algorithm (Hippopotamus and Pufferfish Optimization) with existing methods based on key factors like time consumption, efficiency, and latency in figures 3, 4, and 5. The study evaluates its performance against other optimization techniques, including Artificial Bee Colony, Grasshopper Algorithm, and Particle Swarm Optimization. The findings show that the hybrid approach improves resource utilization, workload distribution, and computational efficiency, ensuring better container resource allocation in cloud computing.

VII. CONCLUSION

This study on container resource allocation in CC demonstrates that hybrid optimization techniques outperform traditional individual optimization methods. While normal optimization approaches yield average results, hybrid techniques achieve superior efficiency, time consumption and latency resulting in significantly better overall performance. By combining the strengths of multiple optimization strategies, hybrid methods offer robust and adaptable solutions for resource allocation challenges in dynamic cloud environments. These findings establish the effectiveness of hybrid techniques in achieving optimal outcomes with reduced computational overhead. Future research can focus on integrating AI-based predictive models, parallel optimization strategies, and adaptive learning techniques to further enhance the resource allocation process. Additionally, incorporating network-aware scheduling can minimize latency and improve overall system responsiveness. These refinements will contribute to developing more intelligent and adaptive cloud computing resource management solutions. The hybrid metaheuristic optimization techniques can address emerging challenges and further improve resource allocation efficiency in CC systems.

REFERENCES

[1] K. D. Foote, ―A Brief History of Cloud Computing, ‖ DATAVERSITY, Dec. 17, 2021.https://www.dataversity.net/brief-history-cloudcomputing/ (accessed Jan. 04, 2023).

[2] A. Gupta, V. Pant, S. Kumar, and P. K. Bansal, "Bank loan prediction system using machine learning," in Proceedings of the 2020 9th International Conference on System Modeling and Advancement in Research Trends, SMART 2020, 2020.

[3] A. Z. Bhat, V. R. Naidu, and B. Singh, "MultimediaCloud for Higher Education Establishments: A Reflection," in Advances in Intelligent Systems and Computing, 2019.

[4] Mehta DK, Patel MB, Dangi A, Patwa N, Patel Z, et al. (2024) Exploring the Efficacy of Natural Language Processing and Supervised Learning in the Classification of Fake News Articles. Advances in Robotic Technology (ART) 2(1): 7-12.

[5] Krishnamoorthy S, Dua A, Gupta S (2021) Role of emerging technologies in future iot-driven healthcare 4.0 technologies: a survey, current challenges and future directions. Journal of Ambient Intelligence and Humanized Computing 14: 361-407.

[6] Yang S, Wang X, Wang X, An L, Zhang G (2020) Highperformance docker integration scheme based on OpenStack 4.

[7] Zeng D, Gu L, Pan S, GuoS (2020) Sensing, Communication and Computation. Software Defined Systems, pp: 17-35.

[8] Virendra Tiwari, Dr.Akhilesh A. Waoo, Balendra Garg (2020), "Study on virtualization technology and its importance in cloud computing environment", IJCRT.

[9]. Almurisi N, Tadisetty S. Cloud-based virtualization environment for iot-based wsn: solutions, approaches and challenges. Journal of Ambient Intelligence and Humanized Computing. 2022 Oct;13(10):4681-703

[10]. Goel G, Tanwar P, Bansal V, Sharma S. The challenges and issues with Virtualization in cloud computing. In: 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI). IEEE; 2021. p. 1334-8.

[11]. Huang J, Yu S, Lai X, Wang H. Achieving big data privacy via hybrid cloud-based data anonymization. IEEE Trans Big Data. 2020;7(1):113-24.

[12]. Liu W. Performance Test and Improvement of Computer Network Virtualization Software in Cloud Computing Environment. Security and Communication Networks. 2022 Aug 30;2022.

[13]. Morabito R. Virtualization on internet of things edge devices with container technologies: A performance evaluation. IEEE Access. 2017;5:8835-50.

[14]. Parast FK, Sindhav C, Nikam S, Yekta HI, Kent KB, Hakak S. Cloud computing security: A survey of ser-vice-based models. ComputSecur. 2022; 114:102580.

[15] Zhou, N., Dufour, F., Bode, V., Zinterhof, P., Hammer, N. J., &Kranzlmüller, D. (2023). Towards Confidential Computing: A Secure Cloud Architecture for Big Data Analytics and AI. In 2023 IEEE 16th International Conference on Cloud Computing (CLOUD) (pp. 293-295). IEEE.

[16] Singh, S.; Singh, P.; Tanwar, S. Energy aware resource allocation via MS-SLnO in cloud data center. Multimed. Tools Appl. 2023, 82, 45541–45563.

[17] Zhong Z, Buyya R (2020) A Cost-Efficient Container Orchestration Strategy in Kubernetes- Based Cloud Computing Infrastructures with Heterogeneous Resources. ACM Trans Internet Techno 20(2):1–24

[18] A. Ibrahim, M. Noshy, H. A. Ali, and M. Badawy, ''PAPSO: A power-aware VM placement technique based on particle swarm optimization,'' IEEE Access, vol. 8, pp. 81747–81764, 2020, doi: 10.1109/ACCESS.2020.2990828

[19] V. Barthwal and M. M. S. Rauthan, ''AntPu: A meta-heuristic approach for energy-efficient and SLA aware management of virtual machines in cloud computing,'' Memetic Comput., vol. 13, no. 1, pp. 91–110, Mar. 2021, doi: 10.1007/s12293-020-00320-7.

[20] S. Gharehpasha, M. Masdari, and A. Jafarian, ''Virtual machine placement in cloud data centers using a hybrid multi-verse optimization algorithm,'' Artif. Intell. Rev., vol. 54, no. 3, pp. 2221–2257, Mar. 2021, doi: 10.1007/s10462-020-09903-9

[21] B. Gomathi, B. Saravana Balaji, V. Krishna Kumar, M. Abouhawwash, S. Aljahdali, M. Masud, and N. Kuchuk, ''Multi-objective optimization of energy aware virtual machine placement in cloud data center,'' Intell. Autom. Soft Comput., vol. 33, no. 3, pp. 1771–1785, 2022, doi: 10.32604/iasc.2022.024052

[22] G. Kanagaraj and G. Subashini, ''Uniform distribution elephant herding optimization (UDEHO) based virtual machine consolidation for energyefficient cloud data centres,'' Automatika, vol. 64, no. 3, pp. 530–540, 2023, doi: 10.1080/00051144.2023.2196116.

[23] Wang, H.; Binti Mansor, N.N.; Mokhlis, H.B. Novel Hybrid Optimization Technique for Solar Photovoltaic Output Prediction Using Improved Hippopotamus Algorithm. Appl. Sci. 2024, 14, 7803. https://doi.org/10.3390/app14177803

[24] Malvi Chalavadi Mamatha and Halugona Chandraiah Sateesh Kumar, "Channel Estimation Using Deep Learning Techniques with Hippopotamus Optimization Algorithm for Millimeter Wave MIMO-OFDM System".International Journal of Intelligent Engineering and Systems, Vol.17, No.6, 2024. DOI: 10.22266/ijies2024.1231.95

[25] Al-Baik, O.; Alomari, S.; Alssayed, O.; Gochhait, S.; Leonova, I.; Dutta, U.; Malik, O.P.; Montazeri, Z.; Dehghani, M. Pufferfish Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems. Biomimetics 2024, 9, 65. https://doi.org/10.3390/biomimetics9020065

[26] Himani Saini, Gopal Singh, Manju Rohil, "Design of Hybrid Metaheuristic Optimization Algorithm for Trust-Aware Privacy Preservation in Cloud Computing". International Journal of Computer Networks and Applications (IJCNA), Volume 10, Issue 6, November – December (2023). DOI: 10.22247/ijcna/2023/223690

[27] Manpreet Kaur and Sarpreet Singh, "A Hybrid Optimization Approach in Cloud Computing Based on Yellow Saddle Goatfish and Particle Swarm Optimization Algorithms for Task Scheduling". International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING, ISSN:2147-67992147-6799 www.ijisae.org. IJISAE, 2024, 12(20s), 08–19

[28] Saurabh Singhal, Nakul Gupta, Parveen Berwal, Quadri Noorulhasan Naveed, Ayodele Lasisi and Anteneh Wogasso Wodajo, "Energy Efficient Resource Allocation in Cloud Environment Using Metaheuristic Algorithm". VOLUME 11, 2023. Digital Object Identifier 10.1109/ACCESS.2023.3330434