Systematic Review on Task Scheduling Mechanisms in Cloud Computing

Sahil* Parveen Sadotra* Pradeep Chouksey* Mayank Chopra* Rabia Koser**
*Department of Computer Science & Informatics**
*Central University of Himachal Pradesh**
*Department of Computer Science & IT***
*Govt. P.G. College Rajouri, UT of JK***
sadotramca2k6@gmail.com

Abstract
Cloud computing has revolutionized resource management in contemporary computing environments. Task scheduling is essential for efficiently allocating and utilizing cloud resources. In cloud environments, tasks are assigned based on availability, capacity, and priority. A comprehensive study of cloud task scheduling approaches, algorithms, and strategies is presented. In this study, we investigate performance, scalability, and resource utilization challenges, advantages, and limitations associated with each method. It also examines recent advancements made in task-scheduling mechanisms. In particular, machine learning and artificial intelligence techniques are incorporated to implement intelligent and dynamic scheduling methods. These algorithms adapt to changing workload conditions, optimize resource usage, and enhance performance. Additionally, cloud computing task scheduling mechanisms are evaluated on the basis of load balancing, energy efficiency, fault tolerance, and compliance. Several task scheduling mechanisms have been proposed and implemented in cloud computing. Scheduling methods include priority-based, load-balancing, metaheuristic, elastic, and machine-learning-based schedules. Although each mechanism has its advantages and limitations, their common goal is to improve system performance and load balancing. Nevertheless, cloud computing task scheduling still faces challenges like the dynamic nature of cloud environments, resource heterogeneity, workload variation, and resource contention. However, task scheduling mechanisms have made significant progress in improving resource allocation.

Keywords: *Artificial Intelligence, Cloud computing, Load balancing, Machine learning , Resource utilization, System performance, Task scheduling.*

Introduction

Efficient task scheduling is of paramount importance in cloud computing because it significantly affects system performance, scalability, and resource use. In this systematic review, we provide a comprehensive analysis and assessment of the various methods utilized for task scheduling in cloud computing environments. The review also explores recent trends, advancements, and future directions in task scheduling mechanisms to identify potential areas of improvement in terms of load balancing, energy efficiency, fault tolerance, and adherence to service-level agreements.

The review emphasizes the importance of precise task scheduling in cloud computing for optimal performance, scalability, and resource utilization. Using machine learning and artificial intelligence techniques, this study explores recent developments and trends in task scheduling mechanisms. A comprehensive

overview of task scheduling mechanisms in cloud computing environments is provided in the comparison and analysis section. The mechanisms include priority-based scheduling, deadline-driven scheduling, genetic algorithms, and ant colony optimization. In cloud computing, task scheduling must consider efficient scheduling, high resource utilization, complex service quality, and low economic cost. However, traditional task scheduling algorithms often fail to meet these objectives. The purpose of this study is to explore new approaches for overcoming these limitations by focusing on cloud-computing task scheduling challenges.

Some challenges associated with cloud computing task scheduling include the following:

1. **Scalability**: Cloud computing environments continue to grow in both users and resources. System performance should not be compromised by handling large workloads efficiently.

2. **Energy Efficiency**: Optimal task scheduling should prioritize energy efficiency by considering power consumption and allocating resources efficiently.

3. **Load Balancing**: It is important to distribute workloads evenly on available resources to avoid overloading some resources and underutilizing others.

4. **Fault tolerance**: Task scheduling mechanisms must be fault tolerant, capable of handling failures, and adaptable to changes in resource availability or system conditions.

Additionally, task scheduling mechanisms must be able to adjust resource allocation in response to evolving workloads and priorities. Further, since cloud computing environments are dynamic and intricate, artificial intelligence and machine learning strategies have emerged as potential solutions to cloud computing task scheduling complications. Cloud computing task scheduling is optimized with machine learning algorithms and artificial intelligence methods.

Adaptive resource allocation is based on current workload patterns and system conditions, analyzed from historical data. It improves scalability, energy efficiency, load balancing, and fault tolerance in cloud computing. Using these approaches, task scheduling can be scalable, energy-efficient, load-balanced, and fault-tolerant.

Methodology

In this study, the methodology was selected based on the need to analyze different task scheduling approaches, algorithms, and strategies in cloud computing environments. Furthermore, the methodology should consider current trends in machine learning and artificial intelligence for efficient and flexible task scheduling. We conducted a systematic literature review to meet these criteria. We searched academic journals, conferences, and online databases for articles published over the past decade. In the search, keywords included "cloud computing task scheduling," "machine learning in task scheduling," "cloud computing task scheduling algorithms," and "dynamic cloud computing task scheduling." Inclusion and exclusion criteria were used to filter out irrelevant articles after identifying relevant keywords.
To be eligible for selection, the articles needed to meet certain criteria. These included being published within the past decade, written in English, and specifically addressing task scheduling algorithms and

approaches within cloud computing environments.

The exclusion criteria included articles that were not related to task scheduling, published in languages other than English, and focused on other aspects of cloud computing unrelated to task scheduling.

A total of 8 articles, which met the specified inclusion criteria, were carefully examined and synthesized to offer a comprehensive overview of various task scheduling methods in cloud computing environments.

Various approaches to task scheduling were identified in the selected articles, including hybrid, multicloud, load balancing, metaheuristic, elastic, and machine learning-based. Incorporating various algorithms and strategies enhances resource allocation. Consideration is given to factors such as workload distribution, resource utilization, response time, energy efficiency, and cost optimization.

Various databases, including Google Scholar, Science Direct, Research Gate and Scopus were utilized alongside search engines to conduct the literature search. An extensive search strategy was implemented to ensure accuracy. We removed irrelevant articles by identifying relevant keywords and applying inclusion and exclusion criteria. This systematic literature review provided a comprehensive overview of different task scheduling approaches in cloud computing environments over the past decade. Over the past ten years, there has been significant development and variation in task scheduling methods within cloud computing environments. There is an increasing focus on efficiency, scalability, and adaptability. Moreover, the analysis revealed that scheduling tasks in cloud computing environments is challenging, due to inter-task dependencies, multiple resources being used, fluctuating workloads, and unpredictable resource changes.

An overview of the different task scheduling mechanisms in cloud computing environments is provided by the data from the selected articles. These articles provide insights into the advantages and disadvantages of different task scheduling approaches, facilitating a comparative analysis. There are also gaps in the selected articles, suggesting areas for further research and improvement. Several task scheduling approaches are available in the field of cloud computing, each with their own strengths and limitations. Current task scheduling algorithms face challenges and limitations that require further research and development. Additionally, these articles illustrated the importance of factoring in response time, energy efficiency, and cost optimization in task scheduling approaches. Task scheduling in cloud computing is a multifaceted and continually evolving field.

Findings and gaps for future work from recent papers:

Table 1 : Findings and gaps of recent published papers

Title	Author	Findings	Future Work
Review of metaheuristic scheduling techniques for cloud computing	Mala Kalra, Sarbjeet Singh[1]	An ant colony optimization method and genetic algorithms are both successful metaheuristic scheduling methods within cloud computing.	Future research will examine energy-conscious scheduling, load distribution, fault resilience, and mixed methodologies, and compare algorithms.
Load Balancing in	Dalia	Research deficiencies are	Developing dynamic and

Cloud Computing: A Comprehensive Review of Algorithms and Future Research Directions.	Abdulkareem Shafiq, N.Z. Jhanjhi, and A. Abdullah [2]	highlighted and potential avenues for intelligent, dynamic algorithms are suggested.	intelligent load balancing algorithms that minimize response time and optimize energy efficiency is the goal of future research in this article.
Intelligent multi-agent reinforcement learning model for resource allocation in cloud computing	Ali Belgacem, Saïd Mahmoudi, and Maria Kihl [3]	The research introduces a sophisticated reinforcement learning framework for cloud computing resource distribution.	IMARM solution implementation on real cloud computing platforms is a future task for this research. Further, the algorithm will be enhanced by incorporating more performance factors.
Hybridization of a metaheuristic algorithm for load balancing in a cloud computing environment	U.K. Jena, P.K. Das, and M.R. Kabat [4]	The paper introduces QMPSO for load balancing in cloud computing. Compared to other optimization algorithms, this algorithm improves performance measures and responds faster.	In future, QMPSO needs to be improved and explored in real-world scenarios.
Fault-aware task scheduling in the cloud using min– min and DBSCAN	S.M.F. D Syed Mustapha and Punit Gupta [5]	To improve resource allocation efficiency and service quality in the cloud, min-min and DBSCAN techniques are combined.	Adding machine learning to improve overall performance and assessing how the algorithm scales and performs in large cloud environments are additional research options.
Multiobjective trust-aware task scheduling algorithm for cloud computing using whale optimization	Sudheer Mangalampalli, Ganesh Reddy Karri, and Utku Kose[6]	By incorporating trust levels into cloud computing task scheduling, this research improves decision-making. In simulations, the methodology outperforms existing approaches using whale optimization.	Machine learning techniques, particularly deep reinforcement learning, could be incorporated into the task scheduling algorithm in the future.
Enhancement of the deadline constraint-based task scheduling mechanism for a cloud environment	Suvendu Chandan Nayak ,Sasmita Parida and Prasant Kumar Pattnaik [7]	The study introduces an optimized scheduling methodology for cloud environments, aiming to maximize resource utilization and acceptance rates. This algorithm improves scheduling efficiency while overcoming current algorithm limitations.	Future research for this study could enhance energy efficiency and cost optimization, investigate intricate scheduling scenarios, verify its effectiveness in practical settings, and incorporate it with other cloud management techniques.
Hierarchical Scheduling Mechanisms for Multilingual Information Resources in Cloud Computing	Yaojun Han and Xuemei Luo [8]	An efficient scheduling framework for multilingual information resources in cloud computing is presented, consisting of four tiers and a three-layer model. Optimizing resource scheduling in the cloud is the goal.	It will be possible to create a schedule algorithm to manage multilingual information resources in cloud computing, as well as to integrate ontology, natural language processing, and translation technologies seamlessly.

Hybrid and Multi-Cloud Scheduling

Using hybrid and multi-cloud scheduling, you can optimize resource allocation and improve scalability by combining on-premises infrastructure and multiple cloud providers. Hybrid and multi-cloud scheduling reduce costs, improve performance, and allow more flexibility in distributing workloads.

It also mitigates the risks associated with relying solely on a single cloud provider. Data security and compliance can also be improved by hybrid and multi-cloud scheduling because organizations can choose the most appropriate cloud provider for their data and applications. Hybrid clouds combine private and public clouds, giving organizations more control over sensitive data while taking advantage of public clouds' scalability and efficiency. Multi-clouds combine different public cloud providers that offer different features and capabilities that can be leveraged based on workload requirements.

By combining hybrid and multi-cloud scheduling, organizations can flexibly adjust their resource allocation in response to varying demand levels. As a result, resources are optimally utilized and costs are efficiently managed. Hybrid and multi-cloud scheduling have numerous benefits. Flexible, improved performance, reduced costs, enhanced data security, and dynamic resource scaling. With these benefits, hybrid and multi-cloud scheduling are valuable approaches for organizations seeking to maximize resource efficiency and task scheduling results. In addition to e-commerce platforms, research institutions, which require access to specialized computing resources, and financial institutions that must comply with strict data security regulations, hybrid and multi-cloud scheduling is used in many industries. Moreover, this strategy enables companies to take advantage of unique capabilities and capabilities provided by different cloud providers, such as geographic locations, specialized services, and cost frameworks.

It is important not to underestimate the challenges of hybrid and multicloud scheduling. Organizations must address various challenges when implementing hybrid and multi-cloud scheduling. Data integration and synchronization across different cloud environments, interoperability between cloud platforms, consistency in performance across distributed systems, and preventing data breaches and unauthorized access across multiple cloud environments are among the challenges. Hybrid and multi-cloud scheduling requires careful planning, coordination, and implementation of appropriate technologies and strategies. Integrating and synchronizing data across multiple cloud platforms requires robust tools. Furthermore, they should establish strong interoperability mechanisms to enable smooth communication and collaboration. To safeguard their data and mitigate the risk of data breaches or unauthorized access, organizations must establish comprehensive security measures, including encryption, access control, and monitoring systems.

Cloud Computing Task Scheduling Mechanisms

Computing tasks are assigned to resources efficiently and effectively in cloud computing environments. User satisfaction, resource utilization, cost efficiency, completion time, and throughput are the primary goals of task scheduling mechanisms. Workflow dependencies, task resource requirements, task priorities, resource availability, and task deadlines affect task scheduling mechanisms in cloud computing environments.

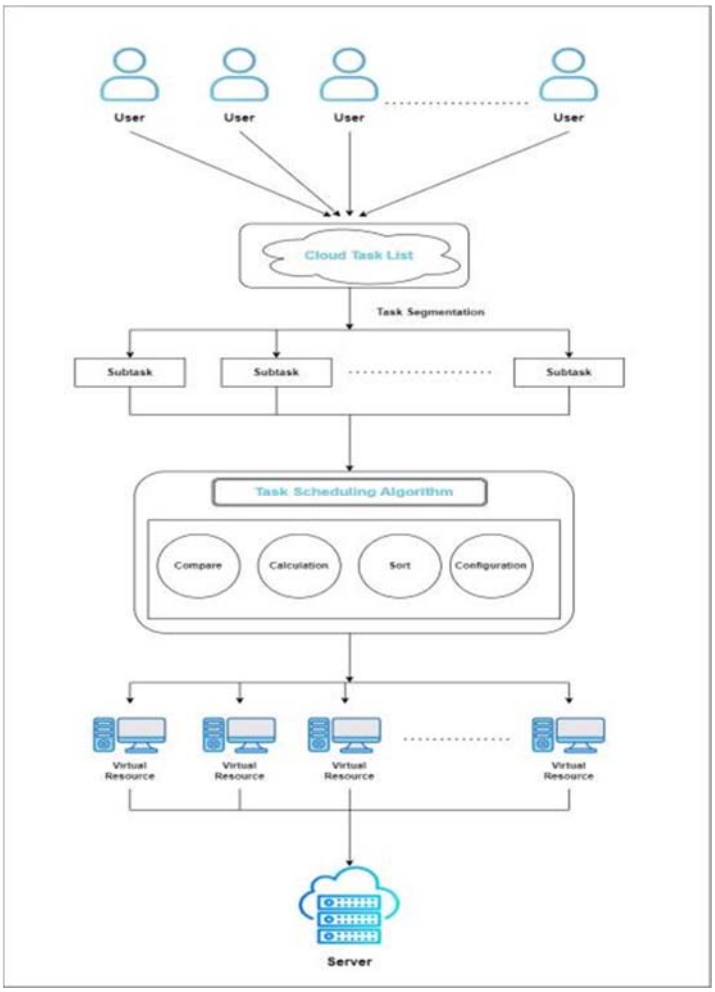

Fig. 1. Task scheduling process in cloud computing

A wide range of task-scheduling mechanisms have been proposed to address these factors, with the most common mechanisms being centralized, decentralized, load balancing, meta-heuristic, elastic, and machine learning-based (discussed in the following sections).

1. Centralized Task Scheduling Mechanisms

Centralized task scheduling mechanisms are based on a global view of cloud computing environments and require a central scheduler. Each cloud computing node sends the tasks and availability status of resources to the central scheduler. Resource selection and task assignment are typically performed in two steps. Resources are selected based on a predetermined set of criteria. Minimal utilization, lowest energy consumption, and maximal user satisfaction are examples of these criteria. The task assignment process allocates selected resources to computing tasks. Cloud computing spaces have been recommended for centralized assignment coordination methods, including Min Min, Max Min, Longest Job First, Shortest Job First, Shortest Job to Machine, space-shared scheduling and backfilling.

As a result, computing task response times, makepan, tardiness, and energy consumption are minimized.

2. Decentralized Task Scheduling Mechanisms

Decentralized task scheduling mechanisms rely on a distributed control model without a central scheduler and require each node in the cloud to manage and allocate resources. Cloud computing nodes maintain local views of the cloud computing environment and allocate resources based on those views without knowing the global view. Therefore, decentralized task-scheduling mechanisms are often more scalable, robust, and resilient to failures than centralized ones. A decentralized task-scheduling mechanism allocates resources to computing tasks using several algorithms in a two-step process. To choose the most appropriate resource, one must consider factors such as its utilization rate, task completion time, and energy consumption. Computing tasks are assigned to selected resources.

Decentralized task scheduling mechanisms include rank-based algorithms, queueing-based mechanisms, and game theory-based mechanisms. Computing tasks are optimized by these algorithms to minimize response and waiting times.

3. Load Balancing Scheduling

Load-balancing scheduling allocates computational tasks across available resources in cloud computing. Through load balancing, resources are used efficiently, throughput is maximized, and response times and latency are minimized. In order to achieve these objectives, several load-balancing scheduling algorithms have been developed. The most common are Round Robin, Weighted Round Robin, and Least Connections. It minimizes average response time, maximizes system throughput, prevents resource bottlenecks, and helps prevent resource overloads by allocating compute tasks to suitable nodes. Two primary types of load-balancing scheduling mechanisms exist in cloud computing: static and dynamic. The static load-balancing method assigns tasks to resources based on criteria such as minimum or maximum number of active processes. A dynamic load-balancing schedule uses a centralized or decentralized mechanism to monitor system resources, including CPU utilization and the number of active processes. By selecting the right resources, these mechanisms balance the system workload. Based on response time, line length, and throughput, the load-balancing scheduling mechanisms were evaluated.

4. Metaheuristic Task Scheduling Mechanisms

Metaheuristic task scheduling mechanisms solve the NP-hard problem of finding an optimal scheduling solution using heuristic search algorithms. System workloads are balanced by evenly distributing computing tasks among the available resources, maximizing system throughput, and reducing response times. Particle Swarm Optimization, Ant Colony Optimization, and Simulated Annealing are well-known metaheuristic task scheduling algorithms used in cloud computing.

Genetic Algorithms and Particle Swarm Optimization are population-based metaheuristic algorithms for finding optimal scheduling solutions. In contrast, Ant Colony Optimization is inspired by ant colonies' collective foraging behavior. Optimal scheduling is achieved by using a probabilistic decision-making strategy. Simulation Annealing uses a probability distribution model to determine the optimal solution. As a result, these metaheuristic task scheduling mechanisms find near-optimal scheduling solutions by distributing computing tasks evenly among the available resources and maximizing the system's capacity.

5. Machine Learning-Based Task Scheduling Mechanisms

Machine learning-based task scheduling mechanisms optimize task scheduling in cloud computing environments using machine learning algorithms and techniques. They optimize task assignments, predict future resource requirements, and adapt to changing workload patterns. A notable advantage of machine-learning-based task scheduling mechanisms is their ability to predict resource requirements based on historical data. Cloud computing environments benefit from more informed and proactive decision making based on past data.

Adaptability and self-adjustment are also advantages of machine-learning-based task scheduling mechanisms. By adapting to changing workload patterns, these mechanisms ensure efficient resource utilization and optimal task completion. Additionally, machine learning-based task scheduling mechanisms can support automated decision-making processes, reduce manual intervention, and minimize human error. Machine-learning-based task scheduling mechanisms require a large amount of high-quality training data, however.

Further, the complexity of creating and implementing machine-learning models for task scheduling can hinder their adoption, requiring both cloud computing and machine learning expertise. Dynamically distributing resources in cloud computing environments involves intricacies and difficulties.

Comparison and Analysis

A range of performance evaluation metrics, as well as scalability and robustness, must be considered when analyzing and evaluating task scheduling algorithms. The effectiveness of various task scheduling methods can be measured by metrics such as makespan, response time, throughput, resource utilization, fairness, and energy consumption. In spite of their importance, these metrics may not fully capture how an algorithm copes with large workloads or adapts to uncertain conditions. Scalability measures the algorithm's ability to handle increasing workloads, whereas robustness evaluates its ability to withstand uncertainty.

FCFS, Round Robin, Shortest Job Next, Priority, Deadline, Least load, Min Min, Max Min, Heuristic, and machine learning-based scheduling are common scheduling algorithms. It is important to consider each algorithm's strengths and limitations when choosing a scheduling approach. Here is a description and analysis of the algorithms

Table 2 : Description and Analysis of different algorithms

Algorithm	Description	Analysis
FCFS	The First-Come, First-Served scheduling algorithm organizes tasks according to their arrival order.	Despite its simplicity, FCFS scheduling algorithm can result in increased waiting times for tasks with long execution periods.
Round Robin	In round-robin scheduling, tasks are assigned a fixed time quantum and executed in cycles, ensuring equal processing opportunities.	Tasks with different execution times may not be efficient. Work on shorter tasks can be delayed and inefficient.
Shortest Job First	It assigns priority to tasks based on their overall execution time, in order to reduce waiting time and maximize system performance.	It minimizes the average waiting time, but it requires knowledge of all tasks' execution times in advance.
Priority Scheduling	Priority scheduling prioritizes tasks by	When high-priority tasks arrive

	importance, so the highest-priority task is executed first.	continuously, priority scheduling can potentially starve low-priority tasks.
Deadline Scheduling	Deadline-based scheduling assigns task deadlines and schedules them based on their deadlines.	It requires accurate estimation and management of task deadlines to ensure timely completion of high priority tasks.
Least Load Scheduling	The least load scheduling algorithm assigns tasks to the resource with the least workload.	A dynamically updated workload or use level may lead to suboptimal task prioritization.
Min Min Scheduling	A Min-Min Scheduling algorithm chooses the task with the shortest execution time and assigns it to the resource with the least workload .	In scenarios where high-priority tasks have tight deadlines or strategic significance, it may make suboptimal scheduling decisions.
Max Min Scheduling	Max Min Scheduling assigns the longest-execution-time tasks to the resource with the least workload.	As it does not directly consider task priority or strategic importance, it may not be suitable for scenarios requiring strict task prioritization or meeting deadlines.
Heuristic Scheduling	An heuristic schedule uses practical rules or guidelines to make scheduling decisions without perfect information.	Heuristic scheduling can be useful when perfect information may not be available or when predicting an optimal solution might be impossible.
Machine Learning-Based Scheduling	In machine learning-based scheduling, decision-making is based on historic data and patterns using machine learning algorithms.	In scenarios with limited or inconsistent historical data, this may not perform well.

Choosing the right task scheduling algorithm requires evaluating the unique requirements and constraints of each scenario. There is no one-size-fits-all approach that guarantees optimal results. Different scheduling algorithms have strengths and limitations to consider. An effective scheduling algorithm considers factors such as task prioritization, resource utilization, information availability, and computational feasibility. By collaborating with stakeholders and continuously evaluating, task scheduling outcomes can be optimized without compromising performance.

Conclusion

The paper examines various approaches to task scheduling in cloud computing, evaluating their advantages and disadvantages.

In cloud computing environments, round-robin, first-come-first-served, shortest-job-next, and priority-based scheduling mechanisms have been examined and implemented. Scheduling tasks has become more complex and advanced as cloud computing expands and becomes more complex.

Hybrid and multi-cloud scheduling mechanisms have been shown to improve resource utilization and performance. Task allocation, response time, and scalability are improved with these scheduling mechanisms. This area has challenges and opportunities for further research. As technology advances and the need for efficient resource usage in cloud computing environments increases, innovative approaches to task scheduling are required. Artificial intelligence and quantum computing should be integrated into such approaches.

Quantum systems have unique properties, such as superposition and entanglement, which can be leveraged in task scheduling algorithms to increase efficiency. In addition, AI can improve overall system performance by enabling more adaptive and intelligent decision making. Further, AI and quantum computing can unleash unmatched computational capabilities in cloud computing by enhancing efficiency and bringing about a remarkable

transformation in this field. There is potential in exploring this area of study to resolve cloud computing scalability, reliability, and effectiveness difficulties. Cloud computing still faces several challenges despite the significant potential demonstrated by quantum computing and artificial intelligence. Scalability and compatibility issues with current cloud computing infrastructure are among the challenges, along with developing strong algorithms to harness quantum computing and artificial intelligence efficiently. Ultimately, quantum computing and artificial intelligence can improve cloud computing task scheduling.

References

[1] M. Kalra and S. Singh, "A Review of Metaheuristic Scheduling Techniques in Cloud Computing," Egyptian Informatics Journal, vol. 16, no. 3, pp. 275–295, Nov. 2015, doi: 10.1016/j.eij.2015.07.001.

[2] D. A. Shafiq, N. Z. Jhanjhi, and A. Abdullah, "Load Balancing Techniques in Cloud Computing environment: a Review," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 7, pp. 3910–3933, Jul. 2022, doi: 10.1016/j.jksuci.2021.02.007.

[3] A. Belgacem, S. Mahmoudi, and M. Kihl, "Intelligent multi-agent Reinforcement Learning Model for Resources Allocation in Cloud Computing," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 6, pp. 2391–2404, Jun. 2022, doi: 10.1016/j.jksuci.2022.03.016.

[4] U. K. Jena, P. K. Das, and M. R. Kabat, "Hybridization of meta-heuristic Algorithm for Load Balancing in Cloud Computing Environment," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 6, pp. 2332–2342, Jun. 2022, doi: 10.1016/j.jksuci.2020.01.012.

[5] S. M. F. D. S. Mustapha and P. Gupta, "Fault Aware Task Scheduling in Cloud Using min-min and DBSCAN," Internet of Things and Cyber-Physical Systems, vol. 4, pp. 68–76, 2024, doi: 10.1016/j.iotcps.2023.07.003.

[6] S. Mangalampalli, G. R. Karri, and U. Kose, "Multi Objective Trust Aware Task Scheduling Algorithm in Cloud Computing Using Whale Optimization," Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 2, pp. 791–809, Feb. 2023, doi: 10.1016/j.jksuci.2023.01.016.

[7] S. C. Nayak, S. Parida, C. Tripathy, and P. K. Pattnaik, "An Enhanced Deadline Constraint Based Task Scheduling Mechanism for Cloud Environment," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 2, pp. 282–294, Feb. 2022, doi: 10.1016/j.jksuci.2018.10.009.

[8] Y. Han and X. Luo, "Hierarchical Scheduling Mechanisms for Multilingual Information Resources in Cloud Computing," AASRI Procedia, vol. 5, pp. 268–273, 2013, doi: 10.1016/j.aasri.2013.10.088.