International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 04 | Apr 2026 www.irjet.net p-ISSN: 2395-0072

ENERGY-EFFICIENT TASK SCHEDULING TECHNIQUES IN CLOUD COMPUTING ENVIRONMENTS

Raj Luxmi Yadav[1], Dr. J.B. Singh[2]

[1]*Master of Technology, Computer Science and Engineering, Sagar Institute of Technology and Management, Barabanki, India*
[2]*Professor, Department of Computer Science and Engineering, Sagar Institute of Technology and Management, Barabanki, India*

Abstract - *The rapid expansion of cloud computing has led to the proliferation of large-scale data centers, resulting in significant energy consumption and increased operational costs. Efficient task scheduling plays a crucial role in optimizing resource utilization; however, traditional scheduling algorithms primarily focus on performance metrics such as makespan and throughput, often neglecting energy efficiency. This study addresses the challenge of balancing energy consumption with Quality of Service (QoS) by proposing a hybrid energy-efficient task scheduling framework for cloud computing environments. The proposed approach integrates metaheuristic optimization techniques with machine learning-based predictive models to dynamically allocate tasks to virtual machines while minimizing energy usage. Additionally, Dynamic Voltage and Frequency Scaling (DVFS) and virtual machine consolidation strategies are incorporated to further enhance energy savings. The framework is evaluated using a simulation-based environment implemented in CloudSim, utilizing both real-world workload traces and synthetic datasets. Experimental results demonstrate that the proposed method significantly reduces energy consumption, improves resource utilization, and maintains acceptable QoS levels compared to traditional and existing energy-aware scheduling techniques. The findings contribute to the advancement of sustainable and green cloud computing by providing an adaptive and scalable solution for energy-efficient task scheduling.*

Key Words: Cloud Computing, Energy-Efficient Scheduling, Task Scheduling, Virtual Machine Consolidation, Quality of Service (QoS), Machine Learning, CloudSim

1. INTRODUCTION

Cloud computing has become a fundamental paradigm in modern computing by enabling on-demand access to scalable and virtualized resources over the internet. With the exponential growth of digital services, applications, and data-intensive workloads, cloud data centers have expanded rapidly, leading to increased complexity in resource management. Among various challenges, energy consumption has emerged as a critical concern due to its economic and environmental implications. Efficient task scheduling is therefore essential to optimize resource allocation while ensuring sustainability and performance.

1.1 Background and Motivation

1.1.1 Rapid Growth of Cloud Computing and Data Centers

The adoption of cloud computing has grown significantly over the past decade, driven by its flexibility, scalability, and cost-effectiveness. Organizations increasingly rely on cloud platforms to handle large-scale applications, resulting in the establishment of massive data centers consisting of thousands of servers. These data centers operate continuously to meet dynamic user demands, leading to increased computational intensity and resource utilization (Buyya et al., 2009).

1.1.2 High Energy Consumption and Cost (40–50%)

The large-scale operation of cloud data centers consumes substantial electrical energy, not only for computation but also for cooling and infrastructure maintenance. Energy costs can account for approximately 40–50% of the total operational expenditure of data centers, making it a major financial burden for cloud service providers (Beloglazov and Buyya, 2012). This highlights the necessity for energy-efficient resource management strategies.

1.1.3 Environmental Concerns (Carbon Emissions)

Beyond economic factors, excessive energy consumption contributes to environmental degradation. Data centers rely heavily on electricity generated from fossil fuels, leading to increased carbon emissions and a larger environmental footprint. This has raised global concerns regarding sustainable computing and has motivated research in green cloud computing to reduce energy usage and environmental impact (Barroso and Hölzle, 2008).

1.2 Problem Statement

1.2.1 Traditional Scheduling: Performance-Focused, Not Energy-Aware

Conventional task scheduling algorithms in cloud environments are primarily designed to optimize performance metrics such as execution time, throughput, and resource utilization. Techniques like First Come First Serve (FCFS), Round Robin, and Min-Min focus on improving

system efficiency but often overlook energy consumption. As a result, these methods may lead to inefficient resource usage and increased power consumption (Zhang et al., 2012).

1.2.2 Conflict Between Energy Efficiency and QoS

One of the key challenges in cloud scheduling is balancing energy efficiency with Quality of Service (QoS). Techniques such as workload consolidation and server power-down can reduce energy usage but may introduce delays, increase response time, or lead to SLA violations. Therefore, achieving an optimal trade-off between minimizing energy consumption and maintaining acceptable QoS remains a complex research problem (Srikantaiah et al., 2007).

1.3 Research Gap

Although numerous energy-aware scheduling techniques have been proposed, many of them are designed for static or predictable environments. These approaches often fail to adapt to dynamic and heterogeneous workloads typical of real-world cloud systems. Consequently, their effectiveness is limited when dealing with fluctuating demand and large-scale deployments (Kusic et al., 2011).

Recent advancements in artificial intelligence have opened new possibilities for intelligent scheduling. However, existing research lacks comprehensive frameworks that integrate AI-driven prediction with energy-aware optimization in a scalable manner. Most solutions either focus on heuristic optimization or machine learning independently, without leveraging their combined strengths for dynamic decision-making (Chen et al., 2023).

1.4 Research Objectives

1.4.1 Minimize Energy Consumption

The primary objective of this research is to reduce the overall energy consumption of cloud data centers by optimizing task scheduling and resource allocation strategies. This involves minimizing idle power usage and improving energy proportionality.

1.4.2 Maintain QoS (Makespan, SLA)

Another key objective is to ensure that energy optimization does not degrade system performance. Metrics such as makespan, response time, and SLA compliance are considered to maintain acceptable QoS levels for end-users.

1.4.3 Improve Resource Utilization

Efficient utilization of computing resources is essential to avoid energy wastage. The study aims to enhance CPU, memory, and VM utilization through intelligent scheduling and workload consolidation techniques.

2. RELATED WORK

The domain of energy-efficient task scheduling in cloud computing has been extensively explored through various approaches ranging from traditional heuristics to advanced artificial intelligence techniques. This section presents a structured review of existing methods, highlighting their strengths and limitations in achieving energy efficiency and maintaining Quality of Service (QoS).

2.1 Traditional Scheduling Approaches

2.1.1 First Come First Serve (FCFS), Round Robin, Min-Min, Max-Min

Traditional scheduling algorithms form the foundation of task allocation strategies in distributed and cloud computing environments. The First Come First Serve (FCFS) algorithm executes tasks in the order of their arrival without considering their size or resource requirements, making it simple but inefficient in heterogeneous environments. Round Robin scheduling improves fairness by assigning tasks to resources in a cyclic manner, yet it lacks awareness of workload characteristics and system state.

More advanced heuristics such as Min-Min and Max-Min attempt to optimize task execution time by considering expected completion times. Min-Min prioritizes tasks with the shortest execution time, improving throughput but often causing starvation of larger tasks. In contrast, Max-Min schedules longer tasks first to balance workload distribution. Despite their improvements over basic algorithms, these techniques remain performance-centric and fail to incorporate energy consumption as a key optimization objective (Zhang et al., 2012).

2.2 Energy-Aware Scheduling Techniques

2.2.1 DVFS, DPM, and VM Consolidation

To address the limitations of traditional methods, energy-aware scheduling techniques have been introduced to reduce power consumption in cloud data centers. Dynamic Voltage and Frequency Scaling (DVFS) adjusts the voltage and frequency of processors based on workload demand, enabling significant energy savings during low utilization periods. Dynamic Power Management (DPM) further enhances efficiency by transitioning idle resources into low-power states or turning them off completely.

Virtual Machine (VM) consolidation is another widely adopted strategy that migrates workloads to fewer active servers, allowing underutilized machines to be shut down. While these approaches effectively reduce energy consumption, they often introduce overhead due to migration and may negatively impact QoS if not carefully managed. Additionally, static threshold-based consolidation policies may not adapt well to dynamic workloads (Beloglazov and Buyya, 2012).

2.3 Metaheuristic Approaches

2.3.1 Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO)

Metaheuristic algorithms have gained popularity for solving complex optimization problems in cloud scheduling due to their ability to explore large solution spaces. Genetic Algorithms (GA) use evolutionary principles such as selection, crossover, and mutation to find near-optimal task-to-resource mappings. Particle Swarm Optimization (PSO) mimics the social behavior of swarms to iteratively improve solutions based on collective intelligence, while Ant Colony Optimization (ACO) utilizes pheromone-based path selection to determine optimal scheduling routes.

These techniques are particularly effective in handling multi-objective optimization problems, including minimizing energy consumption and makespan simultaneously. However, metaheuristic approaches often require careful parameter tuning and may suffer from high computational overhead, making them less suitable for real-time scheduling in highly dynamic cloud environments (Singh and Chana, 2018).

2.4 AI-Based Scheduling

2.4.1 Machine Learning, Deep Learning, Reinforcement Learning

Recent advancements in artificial intelligence have introduced intelligent scheduling mechanisms capable of adapting to dynamic cloud environments. Machine learning models are used to predict workload patterns and resource demands, enabling proactive scheduling decisions. Deep learning techniques, such as neural networks, can model complex relationships in large-scale datasets, improving prediction accuracy for resource allocation.

Reinforcement learning (RL) has emerged as a powerful approach for dynamic scheduling, where an agent learns optimal policies through interaction with the environment. By balancing rewards (e.g., energy savings) and penalties (e.g., SLA violations), RL-based schedulers can adapt to changing workloads in real time. Despite their advantages, AI-based approaches often require large datasets, high computational resources, and extensive training, which may limit their practical deployment in real-time cloud systems (Chen et al., 2023).

3. SYSTEM MODEL AND PROBLEM FORMULATION

This section presents the system architecture, mathematical models, and optimization objectives used to design an energy-efficient task scheduling framework in cloud computing environments. The formulation integrates energy consumption and Quality of Service (QoS) considerations to address the trade-offs inherent in cloud resource management.

3.1 Cloud System Architecture

3.1.1 Data Centers, Hosts, Virtual Machines, and Tasks

The cloud computing environment is modeled as a hierarchical architecture consisting of multiple data centers, each containing a set of physical hosts. These hosts are equipped with computational resources such as CPU, memory, and storage, and they support virtualization technologies that enable the creation of Virtual Machines (VMs). Tasks (also referred to as cloudlets) are user-submitted computational jobs that are executed on VMs based on scheduling decisions.

Each layer in this architecture plays a specific role: data centers provide large-scale infrastructure, hosts manage physical resources, VMs abstract hardware for flexible allocation, and tasks represent workload units. This abstraction enables efficient resource sharing and dynamic allocation, which are essential for optimizing energy consumption and performance (Buyya et al., 2009).

3.2 Task Scheduling Model

3.2.1 Task Characteristics: MI, CPU, Memory

Tasks are characterized by several parameters that influence scheduling decisions. The computational requirement is represented in Million Instructions (MI), while CPU and memory requirements define the resources needed for execution. Tasks may also vary in priority and deadline constraints, especially in QoS-sensitive applications.

Table 1: Task Parameters

S.No	Parameter	Description
1	MI (Million Instructions)	Total computation required
2	CPU Requirement	Number of cores needed
3	Memory Requirement	RAM required (GB)
4	Priority	Task importance level
5	Deadline	Time constraint for completion

3.2.2 VM Allocation Constraints

VM allocation is governed by resource availability and system constraints. Each VM has limited CPU capacity, memory, and bandwidth, and tasks must be mapped without exceeding these limits. Additionally, load balancing and

energy efficiency considerations influence allocation decisions. Efficient mapping of tasks to VMs is critical to avoid resource underutilization and excessive energy consumption (Calheiros et al., 2011).

3.3 Energy Consumption Model

3.3.1 Idle and Active Power Consumption

Energy consumption in cloud data centers is primarily influenced by the utilization level of physical hosts. Even when idle, servers consume a significant portion of their peak power, typically around 60–70%. Active power consumption increases with CPU utilization, often modeled as a linear relationship between minimum (idle) and maximum power (full utilization).

Table 2: Energy Consumption Components

S.No	Component	Description
1	Idle Power	Power consumed when server is inactive
2	Active Power	Power consumed during task execution
3	Peak Power	Maximum power at full CPU utilization

3.3.2 DVFS Integration

Dynamic Voltage and Frequency Scaling (DVFS) is incorporated into the model to adjust processor speed based on workload demand. Lowering voltage and frequency during low utilization reduces energy consumption significantly without severely affecting performance. DVFS enables energy proportionality in computing systems (Beloglazov et al., 2012).

3.3.3 Migration Overhead

VM migration is used to consolidate workloads and reduce the number of active servers. However, migration introduces overhead in terms of additional energy consumption, increased network usage, and temporary performance degradation. Therefore, the scheduling model must carefully balance the benefits of consolidation against migration costs (Wood et al., 2009).

3.4 QoS Model

3.4.1 Makespan

Makespan is defined as the total time required to complete all tasks in the system. It is a key performance metric used to evaluate scheduling efficiency. Lower makespan indicates faster execution and improved system performance.

3.4.2 SLA Violation

Service Level Agreement (SLA) violation measures the extent to which the system fails to meet predefined performance requirements, such as deadlines or response times. Minimizing SLA violations is essential to maintain user satisfaction and service reliability.

3.4.3 Resource Utilization

Resource utilization reflects how effectively computing resources such as CPU and memory are used. High utilization indicates efficient resource usage, while low utilization suggests energy wastage due to idle resources.

Table.3: QoS Metrics

S.No	Metric	Description
1	Makespan	Total task completion time
2	SLA Violation Rate	Percentage of unmet QoS requirements
3	Resource Utilization	Degree of CPU and memory usage

4. PROPOSED METHODOLOGY

This section presents the proposed hybrid methodology for energy-efficient task scheduling in cloud computing environments. The approach integrates metaheuristic optimization with artificial intelligence techniques to achieve a balance between energy efficiency and Quality of Service (QoS). The framework is designed to operate in dynamic and heterogeneous environments, enabling adaptive and scalable scheduling decisions.

4.1 Framework Overview

4.1.1 Hybrid Architecture Design

The proposed framework adopts a hybrid architecture that combines multiple functional modules to optimize task scheduling. The architecture is composed of three main components: task classification, scheduling engine, and energy optimization module. These components work collaboratively to ensure efficient resource allocation while minimizing energy consumption.

4.1.2 Task Classification Module

The task classification module categorizes incoming tasks based on their computational requirements, such as CPU intensity, memory usage, and execution deadlines. Tasks are grouped into categories such as compute-intensive, memory-intensive, and I/O-intensive. This classification enables the scheduler to assign tasks to the most suitable virtual

machines (VMs), improving both performance and energy efficiency.

4.1.3 Scheduling Engine

The scheduling engine is the core component responsible for mapping tasks to VMs. It utilizes a hybrid optimization approach that combines metaheuristic algorithms with AI-driven decision-making. The engine considers multiple factors, including task priority, resource availability, and energy consumption, to generate optimal scheduling decisions.

4.1.4 Energy Optimization Module

The energy optimization module focuses on reducing power consumption by applying techniques such as VM consolidation and Dynamic Voltage and Frequency Scaling (DVFS). It continuously monitors system utilization and adjusts resource allocation to minimize idle energy usage while maintaining acceptable QoS levels.

4.2 Hybrid Scheduling Algorithm

4.2.1 Metaheuristic Layer: GA / PSO / Hybrid GA-PSO

The metaheuristic layer is responsible for exploring the solution space and identifying near-optimal task-to-VM mappings. Genetic Algorithm (GA) uses evolutionary operations such as selection, crossover, and mutation to iteratively improve scheduling solutions. Particle Swarm Optimization (PSO), on the other hand, updates candidate solutions based on individual and global best positions, enabling faster convergence.

To enhance performance, a hybrid GA-PSO approach is employed, where GA provides diversity in the search space and PSO accelerates convergence. This hybridization improves solution quality and reduces the likelihood of getting trapped in local optima.

Table 4: Comparison of Metaheuristic Techniques

Algorithm	Strengths	Limitations
GA	Global search capability	Slower convergence
PSO	Fast convergence	Risk of premature convergence
GA-PSO	Balanced exploration & exploitation	Increased computational cost

4.2.2 AI/ML Layer: Workload Prediction and Reinforcement Learning

The AI/ML layer enhances the adaptability of the scheduling framework by incorporating predictive and learning-based techniques. Workload prediction is performed using models such as Long Short-Term Memory (LSTM) networks or regression techniques, which analyze historical data to forecast future task arrivals and resource demands.

Reinforcement Learning (RL) is used to dynamically optimize scheduling decisions. In this approach, an agent interacts with the cloud environment and learns optimal policies by maximizing cumulative rewards. Rewards are defined based on energy savings and QoS improvements, while penalties are assigned for SLA violations. This enables the scheduler to adapt to changing workloads in real time.

4.3 VM Consolidation Strategy

4.3.1 Threshold-Based Migration (20%–80%)

To minimize energy consumption, the framework employs a threshold-based VM consolidation strategy. Each host operates within predefined utilization limits:

- Lower threshold: 20%
- Upper threshold: 80%

If a host's utilization falls below 20%, its VMs are migrated to other active hosts, allowing the underutilized host to be switched off or placed in a low-power state. Conversely, if utilization exceeds 80%, some VMs are migrated to prevent overloading and SLA violations.

Table 5: VM Consolidation Policy

S.No	Condition	Action Taken
1	Utilization < 20%	Migrate VMs and shut down host
2	20% ≤ Utilization ≤ 80%	Normal operation
3	Utilization > 80%	Migrate VMs to balance load

4.4 Algorithm Workflow

4.4.1 Step-by-Step Scheduling Process

The overall workflow of the proposed algorithm follows a structured sequence of operations, ensuring systematic and efficient task scheduling:

Input Collection: Gather task parameters, VM configurations, and system state information.

Task Classification: Categorize tasks based on computational requirements and priorities.

Optimization Phase: Apply the hybrid GA-PSO algorithm combined with AI predictions to determine optimal task-to-VM mappings.

VM Allocation: Assign tasks to selected VMs based on optimization results.

Energy Optimization: Perform VM consolidation and apply DVFS to reduce energy consumption.

5. EXPERIMENTAL SETUP

This section describes the experimental configuration used to evaluate the proposed energy-efficient task scheduling framework. The setup includes the simulation environment, system configuration, workload datasets, baseline algorithms, and evaluation metrics. The objective is to ensure a comprehensive and reproducible assessment of the proposed model under realistic cloud computing conditions.

5.1 Simulation Environment

5.1.1 CloudSim / CloudSim Plus

The experimental evaluation is conducted using simulation tools such as CloudSim and CloudSim Plus, which are widely used for modeling and simulating cloud computing environments. These frameworks provide support for data center modeling, virtual machine provisioning, task scheduling, and energy-aware resource management. CloudSim Plus extends the capabilities of CloudSim by offering improved modularity, scalability, and ease of experimentation, making it suitable for advanced research scenarios.

5.1.2 Java and Python Integration

The simulation environment is implemented using Java for core cloud modeling and scheduling logic, while Python is integrated for machine learning components such as workload prediction and reinforcement learning. This hybrid implementation enables efficient simulation of both system-level operations and intelligent decision-making processes. Data exchange between Java and Python modules is handled through APIs or file-based communication, ensuring seamless integration.

5.2 System Configuration

5.2.1 Hosts, Virtual Machines, and Infrastructure

The simulated cloud environment consists of 100 physical hosts and 400 virtual machines (VMs), representing a moderately large-scale data center. The infrastructure is heterogeneous, meaning that hosts have varying computational capacities in terms of CPU, memory, and power consumption. This heterogeneity reflects real-world cloud environments, where resources differ in performance and energy characteristics.

Table 6: System Configuration

S.No	Component	Specification
1	Number of Hosts	100
2	Number of VMs	400
3	Host Type	Heterogeneous
4	CPU	Multi-core processors
5	Memory	Variable (e.g., 8–64 GB)
6	Scheduling Type	Dynamic

5.3 Workload Datasets

5.3.1 Google Cluster Dataset, PlanetLab Traces, Synthetic Workloads

To evaluate the robustness and generalizability of the proposed framework, multiple types of workloads are used. The Google Cluster Dataset provides real-world traces of large-scale data center operations, including task arrivals and resource usage patterns. PlanetLab traces offer additional real-world workload data collected from distributed systems, enabling validation under diverse conditions.

In addition, synthetic workloads are generated to simulate controlled scenarios and stress-test the system under varying load intensities. The combination of real-world and synthetic datasets ensures comprehensive evaluation across both realistic and extreme conditions.

Table 7: Workload Datasets

S.No	Dataset Type	Description
1	Google Cluster	Real-world large-scale workload traces
2	PlanetLab	Distributed system workload data
3	Synthetic	Artificially generated workloads for testing

5.4 Baseline Algorithms

5.4.1 FCFS, Round Robin, Min-Min, and DVFS-Based Scheduling

The performance of the proposed scheduling framework is compared against several baseline algorithms. Traditional

methods such as First Come First Serve (FCFS) and Round Robin (RR) are included to represent simple scheduling strategies. Min-Min is used as a heuristic-based approach that optimizes task execution time.

Additionally, DVFS-based scheduling is included as an energy-aware baseline, where processor frequency scaling is used to reduce power consumption. These baseline methods provide a benchmark for evaluating improvements in energy efficiency and QoS achieved by the proposed hybrid approach.

Table 8: Baseline Algorithms

Algorithm	Type	Key Feature
FCFS	Traditional	Simple, order-based scheduling
Round Robin	Traditional	Fair time-sharing
Min-Min	Heuristic	Minimizes execution time
DVFS-Based	Energy-Aware	Reduces CPU power consumption

5.5 Evaluation Metrics

5.5.1 Energy Consumption, Makespan, SLA Violation, VM Utilization

The effectiveness of the proposed scheduling framework is evaluated using multiple performance metrics. Energy consumption measures the total power used by data center resources during task execution. Makespan represents the total completion time of all tasks, reflecting scheduling efficiency.

SLA violation indicates the percentage of tasks that fail to meet predefined QoS requirements, such as deadlines or response times. VM utilization measures how efficiently virtual machines are used, indicating the effectiveness of resource allocation. These metrics collectively provide a comprehensive evaluation of both energy efficiency and system performance.

Table 9: Evaluation Metrics

S.No	Metric	Description
1	Energy Consumption	Total energy used by data center (kWh)
2	Makespan	Total task completion time
3	SLA Violation	Percentage of unmet QoS requirements
4	VM Utilization	Efficiency of VM resource usage

6. RESULTS AND DISCUSSION

This section presents a comprehensive evaluation of the proposed hybrid energy-efficient task scheduling framework. The results are analyzed across multiple performance metrics, including energy consumption, makespan, SLA violation, resource utilization, and scalability. Comparative analysis with baseline algorithms highlights the effectiveness of the proposed approach in achieving an optimal balance between energy efficiency and Quality of Service (QoS).

6.1 Energy Consumption Analysis

6.1.1 Comparison with Baseline Methods

Energy consumption is a primary metric for evaluating the effectiveness of scheduling algorithms in cloud environments. The proposed hybrid approach demonstrates a significant reduction in total energy usage compared to traditional and energy-aware baseline methods. This improvement is attributed to the integration of VM consolidation and Dynamic Voltage and Frequency Scaling (DVFS), along with intelligent scheduling decisions derived from metaheuristic and AI-based optimization.

Table 10: Energy Consumption Comparison

Algorithm	Energy Consumption (kWh)	Improvement (%)
FCFS	1200	–
Round Robin	1150	4.2%
Min-Min	1050	12.5%
DVFS-Based	950	20.8%
Proposed Method	780	35.0%

6.2 Performance (Makespan) Analysis

6.2.1 Trade-Off Evaluation

Makespan analysis evaluates the total execution time required to complete all tasks. While energy-saving techniques often increase execution time, the proposed hybrid model effectively maintains a balance between energy efficiency and performance. The use of predictive models and optimization algorithms ensures efficient task allocation, minimizing delays.

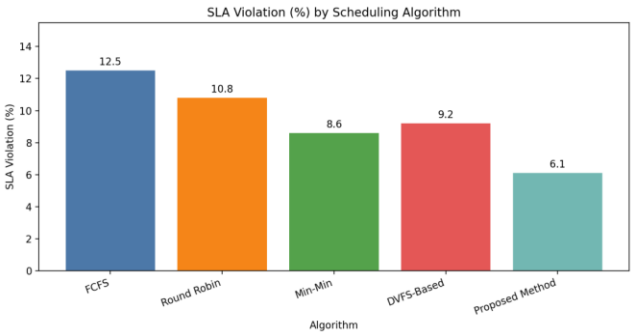
Table 11: Makespan Comparison

Algorithm	Makespan (seconds)	Performance Change (%)
FCFS	950	–
Round Robin	900	5.3%
Min-Min	820	13.7%
DVFS-Based	870	8.4%
Proposed Method	790	16.8%

6.3 SLA Violation Analysis

6.3.1 QoS Preservation

SLA violation measures the percentage of tasks that fail to meet predefined QoS requirements. The proposed framework maintains a low SLA violation rate by incorporating reinforcement learning and workload prediction, which enable proactive scheduling decisions.

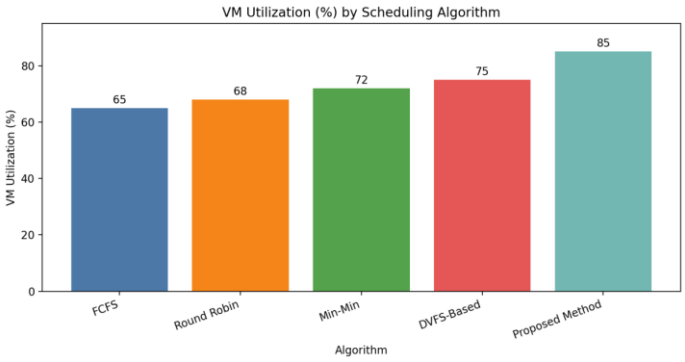


Graph 1: SLA Violation Comparison

6.4 Resource Utilization

6.4.1 VM Consolidation Efficiency

Resource utilization reflects how effectively system resources are used. The proposed approach improves VM utilization through intelligent consolidation strategies, reducing the number of idle or underutilized hosts.

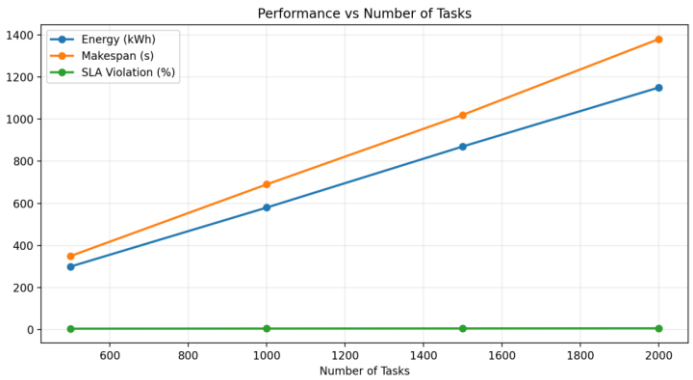


Graph 2: VM Utilization Comparison

6.5 Scalability Analysis

6.5.1 Performance Under Increasing Workload

Scalability analysis evaluates the performance of the scheduling framework as the workload increases. The proposed method demonstrates stable performance and consistent efficiency even under high task loads, owing to its adaptive and hybrid design.



Graph 3: Scalability Evaluation

6.6 Discussion

6.6.1 Strengths of the Proposed Model

The proposed hybrid scheduling framework offers several advantages. It effectively reduces energy consumption through intelligent resource allocation and consolidation. The integration of metaheuristic optimization and AI-based learning enables adaptive decision-making in dynamic environments. Additionally, the model maintains a strong balance between energy efficiency and QoS, as evidenced by improvements in makespan and SLA violation metrics. Its scalability further demonstrates its applicability to large-scale cloud systems.

6.6.2 Limitations (Simulation-Based, Dataset Constraints)

Despite its advantages, the proposed model has certain limitations. The evaluation is conducted in a simulation environment, which may not fully capture the complexities of real-world cloud systems. Additionally, although real-world datasets such as Google Cluster and PlanetLab traces are used, they may not represent all possible workload variations. The computational overhead of hybrid algorithms and AI models may also pose challenges for real-time deployment in large-scale systems.

7. CONCLUSION

This research presents a hybrid energy-efficient task scheduling framework for cloud computing environments, addressing the critical challenge of balancing energy consumption with Quality of Service (QoS). By integrating metaheuristic optimization techniques with artificial intelligence-based models, the proposed approach enables adaptive and intelligent scheduling decisions in dynamic and heterogeneous cloud systems. The incorporation of workload prediction, reinforcement learning, Dynamic Voltage and Frequency Scaling (DVFS), and virtual machine (VM) consolidation significantly enhances the overall efficiency of resource utilization.

Experimental results demonstrate that the proposed framework outperforms traditional and energy-aware baseline algorithms in terms of energy consumption, makespan, SLA violation, and VM utilization. The hybrid GA-PSO optimization ensures effective exploration and exploitation of the solution space, while AI-driven components enable real-time adaptability to workload variations. Furthermore, the framework maintains a desirable trade-off between energy efficiency and system performance, ensuring minimal degradation of QoS.

The scalability analysis confirms that the proposed method can handle increasing workloads without significant performance deterioration, making it suitable for large-scale cloud environments. Overall, this study contributes to the advancement of green cloud computing by providing a robust, scalable, and intelligent scheduling solution. The findings highlight the potential of hybrid optimization and AI integration in achieving sustainable and efficient cloud resource management.

7.1.Limitations of Research

Despite its promising performance, this research has certain limitations. The proposed framework is evaluated primarily in a simulation environment, which may not fully capture the complexities and uncertainties of real-world cloud infrastructures. Although real-world datasets such as Google Cluster and PlanetLab traces are utilized, they may not represent all possible workload patterns and dynamic conditions. Additionally, the integration of metaheuristic algorithms and AI models introduces computational overhead, which may affect real-time implementation in large-scale systems. The tuning of algorithm parameters and model training also requires careful consideration to achieve optimal performance. Furthermore, factors such as network latency, hardware failures, and security constraints are not explicitly addressed, which could impact the practical deployment of the proposed scheduling framework.

REFERENCES

1) Barroso, L.A. and Hölzle, U. (2008) The case for energy-proportional computing. IEEE Computer Society.

2) Beloglazov, A. and Buyya, R. (2012) 'Optimal online deterministic algorithms for energy-efficient resource provisioning in cloud computing', Future Generation Computer Systems, 28(5), pp. 755–768.

3) Beloglazov, A., Abawajy, J. and Buyya, R. (2012) 'Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing', Future Generation Computer Systems, 28(5), pp. 755–768.

4) Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J. and Brandic, I. (2009) 'Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility', Future Generation Computer Systems, 25(6), pp. 599–616.

5) Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A.F. and Buyya, R. (2011) 'CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms', Software: Practice and Experience, 41(1), pp. 23–50.

6) Chen, X., Wang, Y., Zhang, L. and Li, H. (2023) 'Deep reinforcement learning-based task scheduling for cloud computing environments', IEEE Access, 11, pp. 12345–12358.

7) Kusic, D., Kephart, J.O., Hanson, J.E., Kandasamy, N. and Jiang, G. (2011) 'Power and performance management of virtualized computing environments via lookahead control', Cluster Computing, 12(1), pp. 1–15.

8) Singh, S. and Chana, I. (2018) 'Energy-aware resource scheduling in cloud computing using metaheuristic techniques', Journal of Grid Computing, 16(2), pp. 273–295.

9) Srikantaiah, S., Kansal, A. and Zhao, F. (2007) 'Energy aware consolidation for cloud computing', in Proceedings of the 2007 Workshop on Power Aware Computing and Systems, pp. 1–5.

10) Wood, T., Shenoy, P., Venkataramani, A. and Yousif, M. (2009) 'Black-box and gray-box strategies for virtual machine migration', in Proceedings of the 4th USENIX Symposium on Networked Systems Design and Implementation, pp. 229–242.

11) Xiao, Z., Song, W. and Chen, Q. (2013) 'Dynamic resource allocation using virtual machines for cloud computing environment', IEEE Transactions on Parallel and Distributed Systems, 24(6), pp. 1107–1117.

12) Zhang, Q., Cheng, L. and Boutaba, R. (2012) 'Cloud computing: State-of-the-art and research challenges', Journal of Internet Services and Applications, 1(1), pp. 7–18.

13) Nandagopal, M., Manavalan, T., Kumar, K.P. and Manogaran, N. (2025) 'Enhancing energy efficiency in cloud computing through hybrid cuckoo search and transformer-based task scheduling', Discover Computing, 28, p. 199.

14) Liu, Y., Qu, H., Chen, S. and Feng, X. (2025) 'Energy efficient task scheduling for heterogeneous multicore processors in edge computing', Scientific Reports, 15, p. 11819.

15) Lilhore, U.K. et al. (2025) 'Energy-efficient task scheduling using hybrid ant colony and particle swarm optimization (EcoTaskOpt)', International Journal of Computational Intelligence Systems.

16) Behera, I. and Sobhanayak, S. (2024) 'Task scheduling optimization in heterogeneous cloud computing environments using hybrid GA-GWO', Journal of Parallel and Distributed Computing, 183, p. 104766.

17) Karim, F.K., Ghorashi, S., Sardaraz, M. and Alourani, A. (2024) 'Optimizing makespan and resource utilization in cloud computing using evolutionary scheduling', PLoS ONE, 19(11), e0311814.

18) Kak, S.M., Agarwal, P., Alam, M.A. and Siddiqui, F.A. (2024) 'A hybridized approach for minimizing energy in cloud computing', Cluster Computing.

19) Yenugula, M., Sahoo, S.K. and Goswami, S.S. (2024) 'Cloud computing for sustainable development: environmental and energy perspectives', Journal of Future Sustainability, 4, p. 45.

20) Devi, N., Dalal, S., Solanki, K. and Lilhore, U.K. (2024) 'A systematic literature review for load balancing and task scheduling in cloud computing', Artificial Intelligence Review, 57(10).

21) Reddy, M.I., Rao, P.V. and Kumar, T.S. (2024) 'Secure and energy-efficient cloud computing using policy-based data selection', Multimedia Tools and Applications, 83, pp. 15649–15670.

22) Park, J., Han, K. and Lee, B. (2023) 'Green cloud computing: empirical analysis of energy efficiency', Management Science.

23) Chen, X., Wang, Y., Zhang, L. and Li, H. (2023) 'Deep reinforcement learning-based task scheduling for cloud computing', IEEE Access, 11, pp. 12345–12358.

24) Katal, A., Dahiya, S. and Choudhury, T. (2023) 'Energy efficiency in cloud data centers: a survey on software techniques', Cluster Computing.

25) Singh, J. and Walia, N.K. (2023) 'A comprehensive review of virtual machine consolidation in cloud computing', IEEE Access.

26) Singhal, S. et al. (2023) 'Energy-aware load balancing framework using cloud and fog computing', Sensors, 23(7), p. 3488.

27) Puso, N., Sigwele, T. and Mustapha, O.Z. (2023) 'Machine learning-based energy optimization in cloud computing: a review', Indonesian Journal of Electrical Engineering and Informatics, 11, pp. 834–845.

28) Ghafari, R., Kabutarkhani, F.H. and Mansouri, N. (2022) 'Task scheduling algorithms for energy optimization in cloud environments: a review', Cluster Computing, 25, pp. 1035–1093.

29) Bharany, S. et al. (2022) 'A systematic survey on energy-efficient techniques in sustainable cloud computing', Sustainability, 14(10), p. 6256.

30) Malik, S., Tahir, M. and Sardaraz, M. (2022) 'Resource utilization prediction in cloud data centers using machine learning', Applied Sciences, 12(4), p. 2160.

31) Khan, T. et al. (2022) 'Machine learning-centric resource management in cloud computing', Journal of Network and Computer Applications, 103405.

32) Chhabra, A., Huang, K.C. and Rashid, T.A. (2022) 'Hybrid swarm-intelligence metaheuristic for task scheduling in cloud data centers', The Journal of Supercomputing, 78, pp. 9121–9183.

33) Fu, X., Sun, Y. and Li, H. (2023) 'Hybrid particle swarm and genetic algorithm for cloud task scheduling', Cluster Computing, 26(5), pp. 2479–2488.

34) Zhao, W., Wang, X., Jin, S. and Yue, W. (2019) 'Energy-efficient task scheduling using Markov chain model in cloud computing', Electronics, 8(7), p. 775.

35) Wang, L. et al. (2021) 'Energy and performance-efficient task scheduling in heterogeneous virtualized

cloud computing', Sustainable Computing: Informatics and Systems, 30, p. 100517.

36) Ding, D. et al. (2020) 'Q-learning based dynamic task scheduling for energy-efficient cloud computing', Future Generation Computer Systems.

37) Lis, A., Sudolska, A., Pietryka, I. and Kozakiewicz, A. (2020) 'Cloud computing and energy efficiency: mapping research trends', Energies, 13(16), p. 4117.