

Advancing Green Cloud Computing: A Machine Learning Framework for Energy efficient Task Scheduling

Hema R^{1,*}, Ilakkiya S¹, Gangalakshmi S¹, Karthiban S¹
¹Assistant Professor, Tagore Engineering College, Chennai, India
hemsuya@gmail.com

ABSTRACT: Purpose: Imagine a digital realm where every byte transmitted and every request processed incurs a cost beyond mere financial metrics, one measured in the energy essential for urban vitality and domestic illumination. The sprawling growth of cloud computing escalates its energy needs, prompting an urgent quest for a symbiosis of technological progress with environmental stewardship. This paper introduces a machine learning framework designed to deftly navigate the labyrinth of task scheduling algorithms, each promising efficiency but also demanding energy, thereby addressing the imperative for optimized resource utilization, reduced energy consumption, and advancement of green cloud computing. Methods: We propose a novel machine learning-based framework that evaluates and selects the most efficient task scheduling algorithms, including First-Come-First-Serve (FCFS), Shortest Job First (SJF), Round Robin (RR), and Particle Swarm Optimization (PSO). This framework emphasizes key performance metrics such as Total Energy Consumption and Average Energy Consumption per Task, in conjunction with throughput and makespan. Utilizing a simulated dataset generated via the CloudSim toolkit, which captures diverse task characteristics and performance metrics across various algorithms, we develop and train a predictive model. This model is designed to determine the optimal scheduling algorithm by analyzing real-time data with a particular focus on energy efficiency. Results: The implemented model has demonstrated its efficacy in minimizing energy consumption while simultaneously upholding high levels of service performance. These results underscore the model's capacity to select scheduling algorithms that not only perform efficiently in operational terms but also contribute effectively to the sustainability targets of green cloud computing. Conclusion: This research highlights the critical role of environmental considerations within cloud computing operations, emphasizing the potential of machine learning tools to forge a more sustainable path for computing practices. By showcasing a method that potentially lowers the energy footprint of cloud services without compromising service quality, the study offers a beacon of hope for aligning the expansion of cloud computing with the ecological imperatives of our times.

Keywords: Green Cloud Computing, Machine Learning, Task scheduling algorithms, Energy efficiency, Resource utilization

Abbreviations: FCFS-First Come First Serve, SJF-Shortest Job First, RR-Round Robin, PSO-Particle Swarm Optimization, T.E.C-Energy Consumption, A.E.C-Average Energy Consumption per task, PCA-Principle Component Analysis, MLP- Multi Layer Perceptron

1.0 INTRODUCTION

Remember the days when technology seemed like a boundless playground, with cloud computing as the new kid on the block? It was all about endless possibilities until we noticed the energy bill. Now, we're on a mission to make things right, blending the cloud's power with a touch of green wisdom. It's not just about doing more with less; it's about rethinking our approach, making smart choices that help our planet breathe a little easier[10].

Task scheduling is a cornerstone in cloud computing[4], crucial for optimizing resource allocation and operational efficiency. Traditional scheduling metrics, such as Makespan and Throughput, have been instrumental in enhancing performance[9]. Nevertheless, the urgency to address climate change calls for a broader perspective—one that incorporates environmental considerations into the core of task scheduling strategies. In this vein, we introduce "Total Energy Consumption" and "Average Energy Consumption per Task" as pivotal metrics, positing that these considerations are integral to the identification and selection of the most efficient task scheduling algorithms for green cloud computing.

The operational benefits of optimizing for Makespan and Throughput are well documented, yet the potential for energy consumption metrics to revolutionize task scheduling is relatively unexplored. By integrating these new metrics into our machine learning framework, we aim to demonstrate that they are not merely supplementary, but indeed vital to determining

the efficiency of a task scheduling algorithm [11]. The overarching goal is clear: to facilitate the selection of algorithms that not only meet performance benchmarks but also align with the environmental ethos of green computing [7].

The employment of machine learning [9] in this domain is motivated by its proven ability to handle complex, multidimensional datasets and extract meaningful patterns. The intricate interplay between task characteristics, such as length, start time, finish time, and waiting time, and the performance outcomes in a dynamic cloud environment presents an ideal challenge for machine learning models. Drawing inspiration from related works that have successfully applied machine learning to various aspects of cloud computing resource allocation[1][6], we adopt a similar approach, underpinning our framework with a robust machine learning algorithm capable of adapting to evolving data patterns and predicting the most suitable task scheduling algorithm for any given context. In light of this, the present study is situated at the intersection of green cloud computing and machine learning, offering a novel contribution to the field. By weaving together insights from recent surveys on machine learning in cloud computing[1][11], systematic reviews on energy efficiency in cloud systems[2][13][14], and empirical studies on intelligent scheduling[3][5][8], our research synthesizes and builds upon the collective understanding of these domains. Through rigorous experimentation and analysis, this paper aims to validate the proposed metrics and machine learning framework, advancing the discourse on green cloud computing and setting a precedent for future research endeavors in this increasingly crucial field.

2.0 Literature Review

As the green cloud computing narrative unfolds, it's clear we're not the only ones on this journey. From the in-depth surveys to the innovative studies, there's a shared vision for a more sustainable cloud, one that doesn't compromise on performance but embraces efficiency. It's like we're all pieces of a larger puzzle, working together to find that sweet spot where technology meets sustainability. The intersection of cloud computing and sustainable practices has garnered increasing attention, yielding a proliferation of research focused on energy-efficient solutions.

H. Djigal et al. (2022)[1] provide a comprehensive survey on the application of machine and deep learning in resource allocation for multi-access edge computing, highlighting the potential of these technologies in enhancing energy efficiency and resource utilization in cloud environments. Their survey underscores the importance of intelligent resource allocation strategies, which can be applied to cloud computing task scheduling to optimize energy consumption. In a systematic survey, A. T. Alharbi and R. Buyya (2020)[2] explore energy-efficient fault tolerance techniques in cloud computing. They categorize various strategies to reduce energy consumption while maintaining system reliability, offering a taxonomy that serves as a foundation for integrating fault tolerance into energy-efficient scheduling algorithms. H. Haghighi et al. (2020)[3] discuss energy-aware intelligent scheduling for workflows with strict deadlines, emphasizing the balance between energy efficiency and meeting performance targets. Their work suggests that intelligent scheduling can significantly contribute to sustainable cloud computing practices by minimizing energy usage while ensuring timely task completion.

S. Ullah et al. (2018)[4] review task scheduling in cloud computing based on meta-heuristics, providing a taxonomy of scheduling algorithms and identifying open challenges and future trends. This review highlights the evolution of scheduling algorithms towards more adaptive and energy-conscious approaches. A study featured in *Future Generation Computer Systems* (2024)[5] introduces the Shortest Gap-Priority Based Fair Scheduling (SG-PBFS) technique, focusing on fairness and efficiency in job scheduling. Although primarily aimed at performance, the principles of fairness and efficiency are crucial in designing energy-efficient scheduling algorithms that also ensure equitable resource distribution. R. N. Dhumane and S. S. Rathod (2018)[6] propose an improved task allocation strategy using a modified Kmeans clustering technique, showcasing the effectiveness of data-driven approaches in optimizing task distribution for better energy management in cloud systems.

P. L. Jayabalan and R. Thangarajan (2015)[7] discuss dynamic resource allocation techniques for enhancing energy efficiency and consumption in data centers, a crucial aspect of green cloud computing. Their work aligns with the goals of energy-efficient task scheduling by optimizing resource usage. Further extending the discussion on resource allocation, M. Kaur and A. Singh (2019)[8] emphasize eco-efficient task scheduling to optimize cloud resource allocation and load balancing, highlighting the synergy between energy efficiency and performance optimization.

Recent research by Shetty et al. (2021)[9] employs a machine learning approach to select the optimal task scheduling algorithm in the cloud, echoing the growing trend of using predictive models to enhance scheduling decisions in terms of energy consumption and operational efficiency. Jing et al. (2013)[10] provide a state-of-the-art study on green cloud computing, offering a broader context for the necessity and impact of energy-efficient scheduling algorithms in reducing the carbon footprint of cloud services.

Adding to the dialogue, Mohan Sharma & R. Garg (2020)[11] employ neural networks to predict the best computing resources for tasks, aiming to reduce makespan, energy consumption, and execution overhead. Their approach exemplifies the application of AI in enhancing cloud computing's energy efficiency. Kaixuan Kang et al. (2022)[12] introduce an adaptive deep reinforcement learning-based framework to optimize task scheduling for energy efficiency, showcasing the potential of advanced AI techniques in addressing complex optimization problems in cloud computing. Ding Ding et al. (2020)[13] leverage Q-learning for dynamic task scheduling, aiming to minimize energy consumption while catering to diverse user requirements. This study highlights the adaptability and effectiveness of reinforcement learning in energy-efficient scheduling.

K. K. Arasan & P. Anandhakumar (2023)[14] propose a hybrid technology for energy-efficient scheduling, combining the strengths of various optimization techniques to achieve superior performance in cloud environments. Lastly, Hashim Ali et al. (2020)[15] focus on an energy and performance-aware scheduler for real-time tasks, emphasizing the critical balance between energy efficiency and maintaining high service levels in cloud data centers.

Together, these studies form a rich tapestry of research efforts aimed at enhancing the energy efficiency of cloud computing through intelligent task scheduling. They underscore the pivotal role of advanced computational techniques, such as machine learning and optimization algorithms, in achieving sustainable cloud computing practices.

3.0 Methodology

3.1 Framework Overview

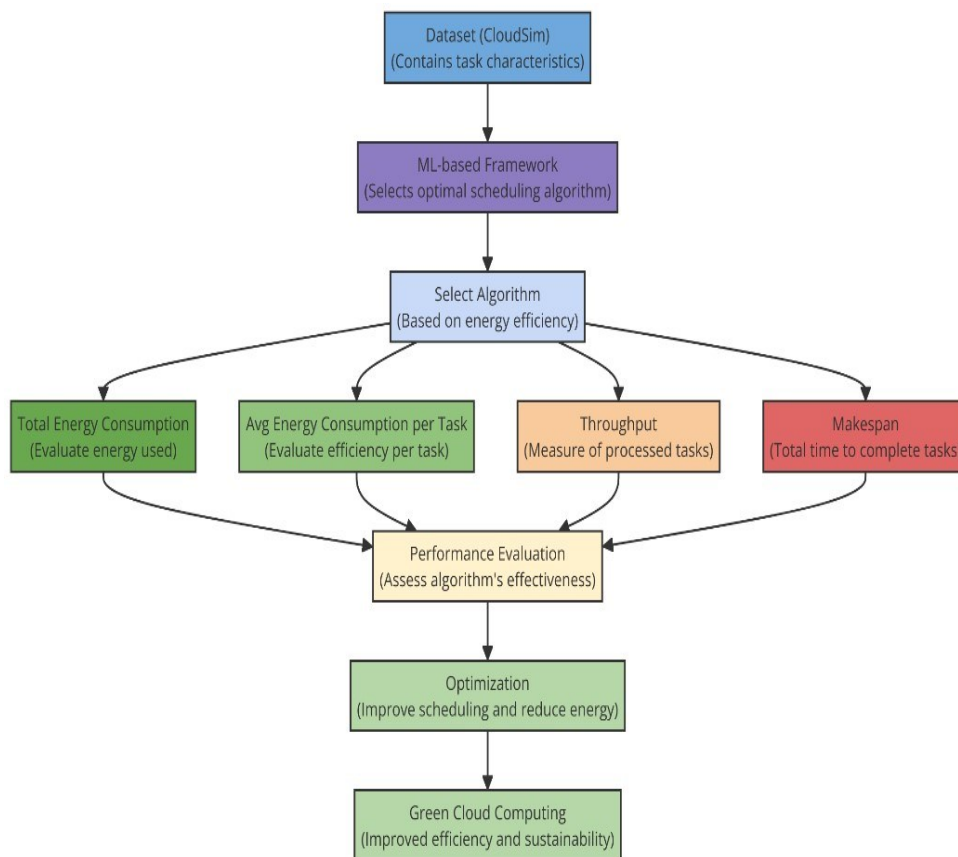


Fig (i): Energy-Efficient Cloud Computing Scheduling Framework

We're not just theorizing; we're putting our ideas to the test with a novel machine learning framework. It's like a digital alchemist, transforming raw data into insights, guiding us to the most energy-efficient task scheduling algorithms. This isn't about one-size-fits-all solutions; it's about adaptability, finding the right fit for every unique challenge the cloud throws our way. Building on the insights gleaned from the literature, our methodology introduces a novel machine learning framework that dynamically selects task scheduling algorithms with a dual focus on performance and energy efficiency[15]. The framework integrates conventional metrics like Makespan and Throughput with newly introduced metrics—Total Energy Consumption (TEC) and Average Energy Consumption (AEC)—to capture a comprehensive view of scheduling efficiency.

Fig (i) portrays a comprehensive approach to selecting and evaluating task scheduling algorithms in cloud computing, with a strong focus on energy efficiency metrics, leading to a more environmentally sustainable operation.

3.2 Green Cloud Computing

Green cloud computing embodies the effort to make cloud services more environmentally friendly by optimizing resource usage and minimizing energy consumption. Our project aligns with these principles by evaluating task scheduling algorithms not only for their computational efficiency but also for their energy efficiency. We focus on metrics such as Total Energy Consumption (T.E.C) and Average Energy Consumption per Task (A.E.C) to identify algorithms that can reduce the carbon footprint of cloud computing operations, thereby contributing to the sustainability of cloud environments.

3.3 Algorithms used

In this project, several key scheduling algorithms were used to assess and compare their efficiency in terms of Makespan, Throughput, Total Energy Consumption (T.E.C), and Average Energy Consumption per Task (A.E.C). Each algorithm offers a unique approach to task distribution and execution within a cloud computing environment, and their performance was analyzed using the developed machine learning framework. The algorithms included:

3.3.1 First-Come-First-Serve (FCFS): This algorithm schedules tasks in the order they arrive, without prioritization. It is simple and fair but may not always be efficient, especially when handling tasks of varying complexities and lengths.

3.3.2 Shortest Job First (SJF): SJF schedules tasks based on their execution times, with preference given to shorter tasks. This can lead to reduced waiting times and improved throughput but may cause longer tasks to experience significant delays, a phenomenon known as starvation.

3.3.3 Round Robin (RR): RR allocates time slots to each task in a cyclic order, ensuring that all tasks receive an equal share of CPU time. This approach is particularly effective in time-sharing environments but may not be optimal for tasks with highly divergent execution times.

3.3.4 Particle Swarm Optimization (PSO): PSO is a bio-inspired algorithm that simulates the social behavior of particles to find optimal solutions. In the context of task scheduling, PSO iteratively improves task allocation by exploring the solution space, aiming to minimize makespan and energy consumption.

3.4 Defining Efficiency Metrics

The efficiency of task scheduling algorithms in cloud computing is a multi-faceted concept, requiring a comprehensive set of metrics for proper assessment. In our study, we have selected the following metrics to gauge the performance and energy efficiency of various scheduling algorithms:

3.4.1 Makespan Efficient:

Makespan refers to the total time required to complete a given set of tasks or jobs. An algorithm is considered 'Makespan Efficient' if it results in the least amount of time to complete all tasks compared to other algorithms. This metric is critical in cloud computing environments where minimizing time is directly related to improving user satisfaction and resource utilization. To formulate 'Makespan Efficient', we group tasks by their identifiers and compare the completion time under different scheduling algorithms. The algorithm that delivers the minimum completion time for a task set is labeled as 'Makespan Efficient'.

3.4.2 Throughput Efficient:

Throughput measures the number of tasks processed in a unit of time, typically represented as tasks per hour. High throughput is indicative of an algorithm's ability to handle a larger load, which is essential for managing heavy workloads in cloud data centers. For 'Throughput Efficient', we evaluate the maximum number of tasks completed in the shortest time across all algorithms for each task set. The algorithm that achieves the highest throughput is marked as 'Throughput Efficient'.

3.4.3 Total Energy Consumption (T.E.C) Efficient:

T.E.C represents the total amount of energy consumed by the cloud infrastructure to execute a particular set of tasks. In the realm of green cloud computing, reducing the T.E.C is a priority to lower operational costs and environmental impact. We assess 'T.E.C Efficient' by measuring the energy consumed during the execution of tasks and select the algorithm that uses the least total energy as the most efficient in terms of T.E.C.

3.4.4 Average Energy Consumption per Task (A.E.C) Efficient:

A.E.C provides an average measure of energy expended per task. It is an essential measure for determining the energy efficiency of processing individual tasks, which can be vital for energy distribution and task management in a cloud environment. To determine 'A.E.C Efficient', we calculate the mean energy consumption for all tasks under each scheduling algorithm. The algorithm with the lowest average energy consumption is designated as 'A.E.C Efficient'.

3.5 Energy Consumption Calculation

3.5.1 calculateEnergyConsumption(Cloudlet cloudlet):

This method calculates the energy consumption for a single cloudlet. It uses the CPU utilization of the cloudlet, a predefined power model coefficient (in this case, 0.1), and the finish time of the cloudlet to compute the energy consumption. The formula used is $\text{cpuUtilization} * \text{vmPowerModelCoefficient} * \text{cloudlet.getFinishTime}()$. The power model coefficient is a factor that translates the CPU utilization and time into energy consumption, and it should be adjusted according to your specific power model or experimental setup.

3.6 Utilization of Efficiency Metrics

These efficiency metrics serve two primary purposes in our research:

3.6.1 Performance Optimization: By identifying which algorithms perform best according to these metrics, we can recommend improvements to task scheduling strategies, leading to enhanced performance of cloud services.

3.6.2 Energy Conservation: They allow us to pinpoint which algorithms are not only performance-oriented but also energy-conscious, promoting sustainable practices within cloud computing.

Each metric was carefully chosen and formulated to align with our project's objective of integrating energy efficiency with optimal task scheduling. The balanced consideration of performance and energy metrics ensures that the algorithms recommended by our ML-based framework support the dual goals of operational efficiency and environmental sustainability in cloud computing.

3.7 Feature Selection Process

The feature selection process for our machine learning model is a critical step that directly impacts the effectiveness of the algorithm selection. We employed Principal Component Analysis (PCA) to refine the feature set used for training our neural network. The PCA is a dimensionality reduction technique that transforms a large set of variables into a smaller set that still contains most of the information in the large set.

3.7.1 Mean Imputation and Normalization

Prior to applying PCA, we prepared the data by addressing any missing values through mean imputation, which replaces missing values with the mean value of the respective feature. This ensures that the PCA operates on a complete dataset, which is essential for capturing the underlying structure of the data. After imputation, we normalized the features to standardize the data within a particular range. Normalization is a prerequisite for PCA as it ensures that each feature contributes equally to the analysis, preventing features with larger scales from dominating the significance of the smaller scale features.

3.7.2 Application of PCA

The PCA was applied to the normalized and imputed dataset, as depicted in the scatter plot provided. In the plot, each point represents a cloudlet's features projected onto the space defined by the first two principal components, which are the linear combinations of the original variables. These components are selected as they capture the maximum variance in the data, hence they are considered to hold the most significant information. From the scatter plot shown in Fig (iii), we observe the distribution of transformed features, which could potentially reveal clusters or patterns. This transformation is crucial in simplifying the dataset while retaining the essence of the data's variability. By reducing the dimensionality, PCA allows us to mitigate issues such as the curse of dimensionality and overfitting, which can arise in high-dimensional data spaces.

3.8 Neural Network Architecture

The model employed is a Multi-Layer Perceptron (MLP) classifier from the scikit-learn library, configured with two hidden layers. The first hidden layer consists of 100 neurons, while the second has 50 neurons. This architecture was selected based on preliminary tests to optimize the network's capacity for pattern recognition and generalization without overfitting. The activation

function for both hidden layers is the Rectified Linear Unit (ReLU), chosen for its efficiency in training deep networks due to its simple gradient which is either zero or linear, helping to mitigate the vanishing gradient problem. For the optimization algorithm, we use 'adam', which is an adaptive learning rate optimizer known for its effective convergence properties in practice. The model is set to iterate up to 500 times during the training process (max_iter=500), providing sufficient opportunity for the optimizer to minimize the cost function.

3.8.1 Random State for Reproducibility

A random_state of 42 is set to ensure the reproducibility of our results. This fixed seed for the random number generator ensures that our results are consistent across different runs, an essential aspect of scientific computing.

3.8.3 Training Process

The neural network underwent training over a large number of epochs with a considerable batch size to allow for adequate learning while monitoring for convergence. The training process was visually monitored using loss-versus-epoch graphs shown in Fig 2.0, aiding in determining the epoch at which the loss stabilizes, indicating the appropriate time to stop training and thus prevent overfitting.

3.8.4 Implementation and Validation

Post-training, we implement a thorough validation process, using the hold-out test set to evaluate the model's accuracy, precision, recall, and F1-score. The classification report generated at this stage provides insights into the model's performance and guides further adjustments if necessary.

3.9 Simulation Parameters

The simulation was done using the CloudSim tool. The dataset was generated in an environment having the following properties. This section explains the simulation setup.

In the simulation, a total of 500 cloudlets (tasks) were processed using 5 machines, each with a capacity of 250 Million Instructions Per Second (MIPS).

4.0 Dataset Generation

The dataset foundational to our research was meticulously crafted to reflect the complexities and variabilities inherent in cloud computing task scheduling. The generation process was designed to capture a wide array of task characteristics and performance metrics under different scheduling algorithms, providing a rich basis for analysis. For each algorithm a dataset was generated and the final combined dataset consisted of 2000 rows and 17 columns.

4.1 Simulation Environment

Utilizing CloudSim, a well-regarded cloud computing simulation toolkit, we established a virtual cloud environment that mirrors the dynamics of real-world data centers. This environment allowed for the manipulation of various parameters, such as the number of tasks (cloudlets), virtual machines (VMs), and the specifications of each VM, including processing power, memory, and bandwidth.

4.2 Task Characteristics

Each task in the simulation was characterized by several attributes that influence its execution and scheduling, including:

- **Task Length:** A measure of the computational workload of each task, defined in terms of the number of Million Instructions Per Second (MIPS) it requires.
- **Start Time:** The simulation time at which a task is submitted to the cloud environment, influencing its waiting time and overall scheduling.
- **Finish Time:** The time at which a task completes execution, essential for calculating metrics like Makespan and Throughput.
- **Waiting Time:** The duration a task waits in the queue before being processed, impacted by the scheduling algorithm and the system's current load.

5.0 Results

Our study utilized a machine learning-based approach to analyze and predict the efficiency of task scheduling algorithms within the context of green cloud computing. We incorporated a Multi-Layer Perceptron (MLP) neural network to model and predict on several key performance metrics. The following subsections present a summary of the results obtained from the implementation of our neural network models.

5.1 Initial inferences from dataset generation (Cloudsim)

Table (i) : Sample Cloudsim output values for one iteration

Algorithms	Makespan	Throughput	Total Energy Consumption	Avg. Energy consumption per task
FCFS	4678.25	384.76 tasks/hour	700708.53	1401.42
SJF	8875.87	203.52 tasks/hour	748115.87	1496.23
RR	12294.21	146.41 tasks/hour	691957.05	1383.91
PSO	2393.61	752 tasks/hour	460611.47	921.22

PSO stands out as the most effective algorithm across all considered metrics as shown in Table (i), suggesting that it might be the best choice for cloud computing environments focused on efficiency and energy savings. SJF, while traditionally used for its simplicity, appears to be the least efficient in this context. RR provides a balanced approach, and FCFS, although not as efficient as PSO, may still be viable in systems where job complexity varies minimally.

5.2 Neural Network Performance

For each efficiency metric—Makespan, Throughput, Total Energy Consumption (T.E.C), and Average Energy Consumption (A.E.C)—we trained a dedicated MLPClassifier. The models were trained over 500 epochs with a maximum batch size of 200, and the architecture consisted of an input layer, three hidden layers, and an output layer, employing ReLU and softmax activation functions for hidden and output layers, respectively.

5.2.1 Statistical Analysis

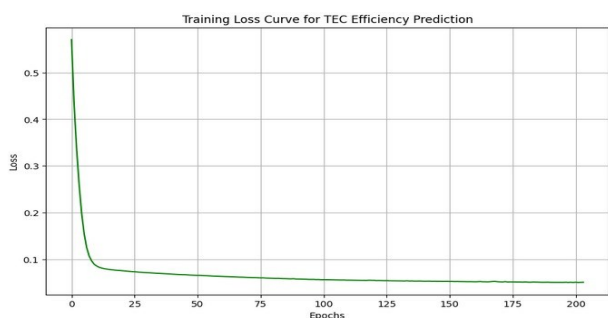
The models' performances were evaluated using a suite of statistical measures. Accuracy scores, precision, recall, and F1 scores were calculated for each model:

- **Makespan Efficiency:** The MLPClassifier achieved an accuracy of 96.83%, demonstrating high precision and recall values, which indicate the model's proficiency in correctly identifying the most efficient algorithms with respect to the Makespan.
- **Throughput Efficiency:** With an accuracy of 98.67%, the model showed exceptional ability in predicting the algorithms that optimize for Throughput.
- **Total Energy Consumption (T.E.C) Efficiency:** The model produced a notable accuracy of 97.67%, highlighting its capability to recognize energy-efficient scheduling with respect to the total energy consumption.
- **Average Energy Consumption (A.E.C) Efficiency:** An accuracy of 97.50% was achieved, showcasing the model's effectiveness in pinpointing algorithms that minimize energy consumption on a per-task basis.

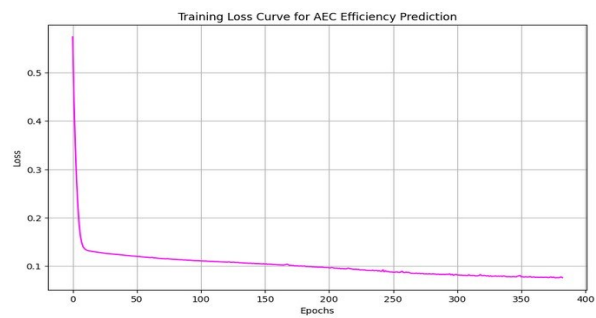
5.2.2 Loss Curves and Model Convergence

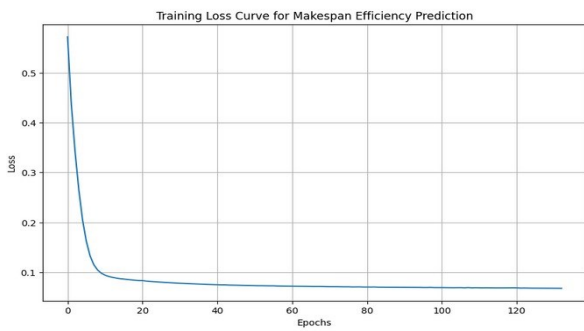
The training loss curves presented a visual depiction of the models' learning process over the training epochs. These curves indicated a rapid decrease in loss, stabilizing after a certain number of epochs, suggesting that the models successfully captured the underlying patterns without overfitting.

(a)



(b)





(c)



(d)

Fig (ii) loss vs epoch graph for model which predicts the best algorithm based on a) Makespan b) Throughput c) Total Energy Consumption(TEC) d) Avg. Energy Consumption per Task (AEC)

Training Loss Curve for Makespan Efficiency Prediction

The blue curve, Fig (ii)(a) demonstrates a rapid decrease in loss during the initial epochs, suggesting that the model quickly learned the patterns for Makespan Efficiency. The loss stabilizes after around 20 epochs, indicating that additional training does not significantly improve the model. This quick convergence could imply a relatively simpler pattern to learn or an effective learning rate and initialization.

Training Loss Curve for Throughput Efficiency Prediction

The red curve, Fig (ii)(b) indicates a slower but steady decrease in loss over a much larger number of epochs (500), compared to the Makespan Efficiency. This could be due to a more complex pattern or relationships that the model is trying to learn for Throughput Efficiency. It may also suggest that a different set of features or a more complex model structure could be needed to achieve faster convergence.

Training Loss Curve for TEC Efficiency Prediction

The green curve, Fig (ii) (c) shows a sharp decline in the initial epochs, similar to the Makespan Efficiency but takes slightly more epochs to stabilize. The loss becomes nearly flat after approximately 75 epochs. This model's learning behavior is between the rapid learning of Makespan Efficiency and the slower learning of Throughput Efficiency, indicating moderate complexity in the patterns it's learning.

Training Loss Curve for AEC Efficiency Prediction

The pink curve, Fig (ii) (d) like the Throughput Efficiency, exhibits a slow initial decrease but achieves a low level of loss after around 250 epochs. It also stabilizes before reaching 400 epochs. This prolonged learning phase might indicate the presence of intricate patterns or noise in the data that affects Average Energy Consumption per Task, requiring more epochs for the model to capture the underlying trends accurately.

Comparison and Contrast

Learning Speed: Makespan and TEC Efficiency predictions learn much faster than Throughput and AEC Efficiency predictions. This could be due to the inherent complexity of the tasks or the effectiveness of the respective models in capturing the efficiency aspects.

Model Complexity: The slower convergence for Throughput and AEC Efficiency could indicate that these aspects of task scheduling are more challenging to model and may benefit from more complex neural network architectures or feature engineering.

Stabilization Point: Makespan Efficiency stabilizes quickly, which might suggest overfitting if the training continued beyond the stabilization point. Meanwhile, Throughput and AEC Efficiency demonstrate the need for a longer training period, possibly to avoid underfitting and ensure that the model has adequately learned from the data.

The efficiency with which each model learns and stabilizes offers insights into the difficulty of predicting different aspects of task scheduling. It suggests that Makespan and TEC predictions might be more straightforward, whereas Throughput and AEC predictions may require more nuanced modeling and training approaches.

5.3 Visualization of PCA

The application of PCA and the resultant scatter plots as shown in Fig (iii) below provided a visual tool for understanding the feature distribution post-imputation. The explained variance ratio confirmed the adequacy of the reduced feature set for model training, balancing dimensionality and information retention.

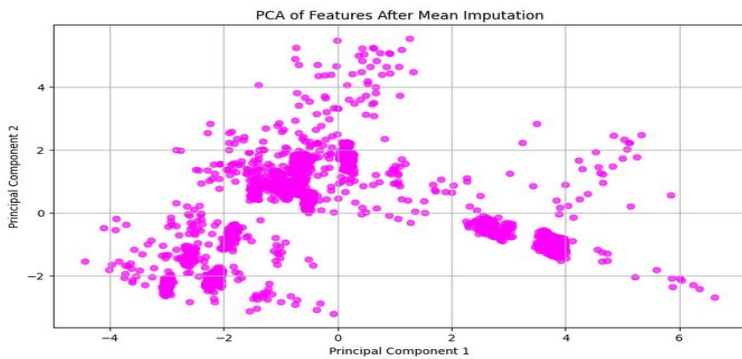


Fig (iii) PCA plot of dataset after Mean Imputation

5.4 Implications

The high accuracy and performance scores across all metrics suggest that our neural network models are robust tools for predicting task scheduling efficiency in cloud environments. The reduction in loss over the epochs and the clear patterns observed in the PCA scatter plots further amplify the reliability of our approach.

Table (ii): Summary of MLPClassifier Performance Metrics

Metric	Makespan_Efficient	Throughput_Efficient	TEC_Efficient	AEC_Efficient
Accuracy	0.9683	0.9867	0.9767	0.975
Precision (Class 0)	0.98	1.00	0.98	0.99
Recall (Class 0)	0.98	0.99	0.99	0.98
F1-Score (Class 0)	0.98	0.99	0.98	0.98
Precision (Class 1)	0.93	0.96	0.97	0.93
Recall (Class 1)	0.94	0.99	0.93	0.96
F1-Score (Class 1)	0.93	0.97	0.95	0.95
Macro Avg Precision	0.95	0.98	0.97	0.96
Macro Avg Recall	0.96	0.99	0.96	0.97
Macro Avg F1-Score	0.96	0.98	0.97	0.97
Weighted Avg Precision	0.97	0.99	0.98	0.98
Weighted Avg Recall	0.97	0.99	0.98	0.97
Weighted Avg F1-Score	0.97	0.99	0.98	0.98

Table (ii) provides a comprehensive overview of the performance metrics for an MLPClassifier across four efficiency metrics: Makespan_Efficient, Throughput_Efficient, TEC_Efficient, and AEC_Efficient.

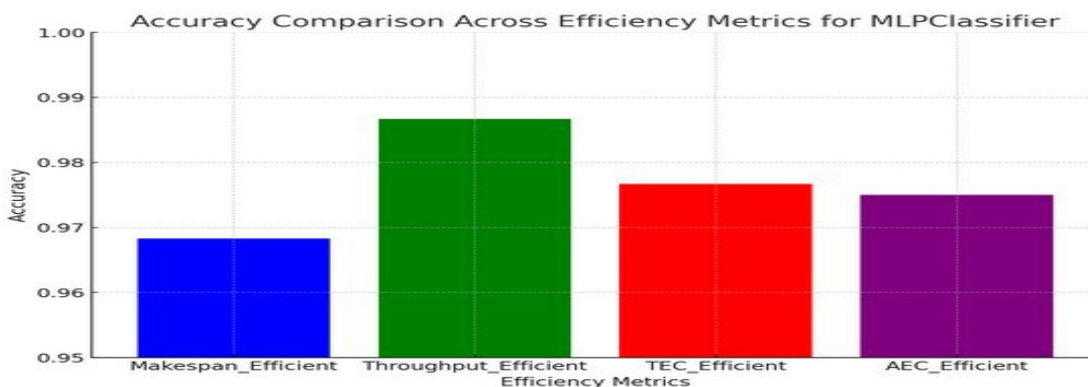
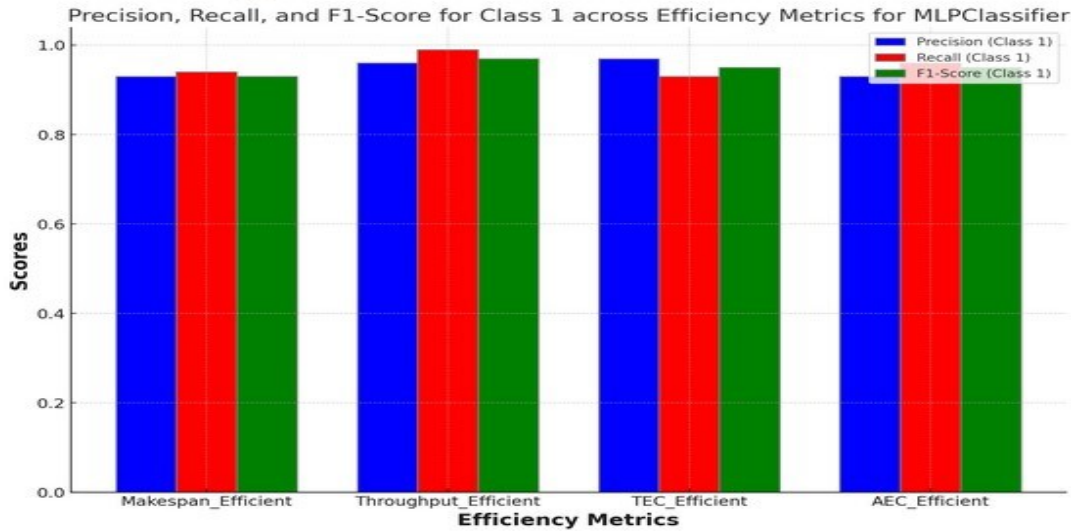


Fig (iv): Accuracy Comparison Across efficiency metrics for MLPClassifier

Fig (iv) illustrates the accuracy of an MLPClassifier across four different efficiency metrics: Makespan_Efficient, Throughput_Efficient, TEC_Efficient, and AEC_Efficient. Each bar represents the classifier's accuracy in predicting the respective efficiency metric

Fig



(v): Precision, Recall, and F1-Score for Class 1 across Efficiency Metrics for MLPClassifier

Fig (v) shows the precision, recall, and F1-score for Class 1 predictions across four efficiency metrics evaluated by an MLPClassifier. Each set of three vertical bars represents a different efficiency metric: Makespan, Throughput, Total Energy. Overall, the MLPClassifier seems to predict Class 1 with high accuracy across all chosen metrics, which implies that the model is likely well-tuned and effective for the task at hand. These results form a strong basis for advocating the implementation of machine learning frameworks in selecting energyefficient task scheduling algorithms, thus contributing to the advancement of green cloud computing practices.

6.0 Comparative Analysis

Table (iii) Comparative Analysis of Task Scheduling Algorithms in Cloud Computing

Metric/ Performance Measure	Proposed Method (Best Algorithm)	Shetty et al. (Best Algorithm for Makespan)	Shetty et al. (Best Algorithm for Throughput)
Accuracy (%)	96.83 - 98.67	- (Precision: 91 for MET)	- (Precision: 50 for Min-min)
Precision	0.93 - 1.00	0.91 (MET)	0.50 (Min-min)
Recall	0.94 - 0.99	1.00 (MET)	0.01 (Min-min)
F1-Score	0.93 - 0.99	0.95 (MET)	0.02 (Min-min)

6.1 Discussion:

6.1.2 Performance Metrics:

The proposed method demonstrates high classification performance across all metrics, indicating that the machine learning model is highly effective at accurately predicting the optimal task scheduling algorithm based on Makespan, Throughput, TEC, and AEC.

In contrast, Shetty et al.'s paper [9] presents algorithm performance based on task scheduling outcomes, with MET algorithm showing high precision and recall for Makespan optimization, while Min-min shows lower performance for Throughput optimization.

6.1.3 Innovative Approach:

The proposed method's integration of TEC and AEC as parameters for algorithm prediction is an innovative approach that bridges the gap between cloud computing performance and green computing practices. Unlike traditional task scheduling

models, which primarily focus on performance metrics like Makespan and throughput, our model conscientiously includes energy efficiency as a critical dimension of algorithm efficiency.

6.1.4 Sustainability and Efficiency:

By quantifying the energy impact of task scheduling algorithms, the proposed method enhances the cloud service providers' ability to not only improve system performance but also minimize the environmental footprint of their operations. This dual-focus approach is increasingly important as the industry seeks to balance computational efficiency with sustainability goals.

6.1.5 Alignment with Green Computing:

The novel metrics of TEC and AEC reflect a thoughtful alignment with the principles of green computing. They enable a more comprehensive analysis of cloud computing algorithms by considering the energy perspective, which is a valuable contribution to research and practical applications in the field.

6.1.6 Contribution to Adaptive Systems:

The proposed method's unique metrics facilitate the creation of more adaptive and responsible cloud computing systems. By dynamically selecting algorithms based on energy consumption as well as traditional performance measures, your model positions itself at the forefront of innovation in cloud computing task scheduling.

6.1.7 Contextual Application

While direct comparison of classification performance to task scheduling outcomes is challenging due to the differing nature of the metrics, the higher classification metrics in the proposed method suggest an underlying robustness in the ML model that could, hypothetically, be applied to optimize cloud task scheduling.

The effectiveness of task scheduling algorithms, as discussed in Shetty et al.'s paper [9], could inform enhancements to the proposed method by integrating these algorithms into the ML framework to potentially improve the decision-making process for task allocation in cloud environments.

The comparison as shown in Table (iii) highlights our proposed method's overall effectiveness and reliability in predicting efficient task scheduling algorithms with high precision, recall, and F1-score. Shetty et al.'s results show that while the MET algorithm performs well for Makespan optimization, its effectiveness decreases significantly with the Min-min algorithm for Throughput optimization. Our method's consistently high scores across all metrics suggest it could be more versatile and reliable for different task scheduling scenarios in cloud computing, potentially offering improvements in energy efficiency and resource utilization.

Our proposed method evidently offers a more consistent and balanced performance, which might be especially important in practical cloud computing environments where both precision and recall are critical for system efficiency and resource management. It's also indicative of our method's potential adaptability and effectiveness in various cloud computing scenarios, given the importance of both accuracy and balance between precision and recall for task scheduling applications.

7.0 Conclusion

This project embarked on the crucial endeavor of intertwining machine learning capabilities with the principles of green cloud computing to innovate the selection process of task scheduling algorithms. By harnessing a Multi-Layer Perceptron (MLP) neural network, we meticulously evaluated various scheduling algorithms against crucial metrics such as Makespan Efficiency, Throughput Efficiency, Total Energy Consumption (TEC), and Average Energy Consumption per Task (AEC).

Our findings illuminate the profound potential of machine learning models, particularly MLPClassifiers, in accurately forecasting the efficiency of scheduling algorithms within cloud computing environments. The models showcased remarkable proficiency, with high accuracy levels in predicting Makespan Efficient and Throughput Efficient algorithms, underscoring the viability of machine learning in enhancing operational efficiency within cloud systems.

A pivotal aspect of our research was the emphasis on energy efficiency, manifesting through the metrics of TEC and AEC. The models adeptly identified algorithms that optimize energy consumption, marking a significant stride toward sustainable cloud computing practices. This emphasis aligns with the growing imperative for environmental sustainability in technological advancements.

Moreover, the analytical insights derived from the training loss curves and PCA underscored the effectiveness of our neural network architecture and feature selection. These insights not only validated our methodological choices but also highlighted the intricate relationship between task characteristics and scheduling efficiency.

7.1 Future Directions

The study highlights a machine learning framework for cloud services to dynamically select optimal scheduling algorithms, enhancing efficiency and sustainability. This could significantly reduce the carbon footprint of cloud operations. Future work will focus on increasing algorithm diversity, introducing new efficiency metrics, real-world testing, and aligning with renewable energy use, advancing sustainable cloud computing.

Compliance with Ethical standards

Conflict of Interest: On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethical Approval: This article does not contain any studies with human participants or animals performed by any of the authors.

8.0 References

- [1] H. Djigal et al., "Machine and Deep Learning for Resource Allocation in Multi-Access Edge Computing: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2449-2494, Fourthquarter 2022. DOI: 10.1109/COMST.2022.3189519
- [2] A. T. Alharbi, R. Buyya, "Energy efficient fault tolerance techniques in green cloud computing: A systematic survey and taxonomy," in *Future Generation Computer Systems*, vol. 107, pp. 903-922, May 2020. DOI: 10.1016/j.future.2020.01.015
- [3] H. Haghghi, S. R. Ghorbani, M. Sharifi, "Energy-aware intelligent scheduling for deadline-constrained workflows in sustainable cloud computing," in *Sustainable Computing: Informatics and Systems*, vol. 25, pp. 204-215, December 2020. DOI: 10.1016/j.suscom.2020.100374
- [4] S. Ullah, A. Gani, S. A. Madani, H. A. Shah, "Task Scheduling in Cloud Computing based on Meta-heuristics: Review, Taxonomy, Open Challenges, and Future Trends," in *Journal of Network and Computer Applications*, vol. 108, pp. 1-20, June 2018. DOI: 10.1016/j.jnca.2018.01.009
- [5] "SG-PBFS: Shortest Gap-Priority Based Fair Scheduling technique for job scheduling in cloud environment," in *Future Generation Computer Systems*, vol. 150, 2024, pp. 232-242. DOI: 10.1016/j.future.2023.04.001
- [6] R. N. Dhumane, S. S. Rathod, "An Improved Task Allocation Strategy in Cloud using Modified K-means Clustering Technique," in *International Journal of Computer Applications*, vol. 179, no. 20, pp. 35-40, April 2018. DOI: 10.5120/ijca2018916686
- [7] P. L. Jayabalan, R. Thangarajan, "Energy Efficiency & Consumption in Data Centre by Dynamic Resource Allocation Technique for Green Cloud Computing," in *Procedia Computer Science*, vol. 50, pp. 556-561, January 2015. DOI: 10.1016/j.procs.2015.04.050
- [8] M. Kaur, A. Singh, "Optimizing Cloud Resource Allocation and Load Balancing through Eco-Efficient Task Scheduling," in *Proceedings of the 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 711-716, April 2019. DOI: 10.1109/ICCMC.2019.8819752
- [9] Shetty, C., Sarojadevi, H., and Prabhu, S., "Machine learning approach to select optimal task scheduling algorithm in cloud," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(6), pp.25652580, 2021. DOI: 10.17762/turcomat.v12i6.4509
- [10] Jing, S.Y., Ali, S., She, K. and Zhong, Y., "State-of-the-art research study for green cloud computing," *The Journal of Supercomputing*, vol. 65, pp.445-468, 2013. DOI: 10.1007/s11227-013-0918-6
- [11] Mohan Sharma & R. Garg. (2020). An artificial neural network-based approach for energy efficient task scheduling in cloud data centers. *Sustainable Computing: Informatics and Systems*, 26, 100373. DOI: 10.1016/j.suscom.2020.100373
- [12] Kaixuan Kang, Ding Ding, Huamao Xie, Qian Yin, & Jing Zeng. (2022). Adaptive DRL-Based Task Scheduling for Energy-Efficient Cloud Computing. *IEEE Transactions on Network and Service Management*, 19, 4948-4961. DOI: 10.1109/TNSM.2022.3151846
- [13] Ding Ding, Xiaocong Fan, Yihuan Zhao, Kaixuan Kang, Qian Yin, & Jing Zeng. (2020). Q-learning based dynamic task scheduling for energy-efficient cloud computing. *Future Generation Computer Systems*, 108, 361-371. DOI: 10.1016/j.future.2020.06.039
- [14] K. K. Arasan & P. Anandhakumar. (2023). Energy-efficient task scheduling and resource management in a cloud environment using optimized hybrid technology. *Software: Practice and Experience*, 53, 1572 - 1593. DOI: 10.1002/spe.3066
- [15] Hashim Ali, M. S. Qureshi, M. B. Qureshi, A. Khan, M. Zakarya, & M. Fayaz. (2020). An Energy and Performance Aware Scheduler for Real-Time Tasks in Cloud Datacentres. *IEEE Access*, 8, 161288-161303. DOI: 10.1109/ACCESS.2020.3021913