Task Scheduling Algorithm for Cloud Computing Using Hybrid Swarm Intelligence

FARHAT KHAN[1], PROF. MAHENDRA SAHARE[2], PROF. VINOD YADAV[3]
[1]*M. Tech Scholar, CSE Department, NIRT Bhopal*
[2]*Associate Professor, CSE Department, NIIST Bhopal*
[3]*Associate Professor, CSE Department, NIRT Bhopal*

Abstract— Cloud computing is a popular technology that allows customers to use computing resources remotely on a pay-as-you-go basis. Task scheduling is a significant challenge in cloud computing environments, as tasks must be scheduled effectively to reduce implementation time and cost while optimizing resource efficiency. This research develops and evaluates a hybrid swarm intelligence method for multi-objective task scheduling in cloud computing, combining an estimate of the distribution algorithm (EDA) with particle swarm optimization (PSO) and ant colony optimization (ACO). Most existing methods do not fully leverage EDA, leading to high task completion times. This work focuses on effectively integrating EDA with the firefly algorithm to reduce task completion times in scheduling algorithms. The findings indicate that the proposed algorithm excels in terms of time efficiency and faster convergence compared to existing methods. Moreover, for both small and large-scale activities, the proposed algorithm demonstrates greater efficiency with a makespan of 1000.74 seconds, throughput of 64.30%, and resource utilization of 99.90%.

Index Terms- Cloud Computing, Task Scheduling, Virtual Machine, Swarm Intelligence.

I. INTRODUCTION

Cloud computing is an open platform for on-demand access to services. The stack of cloud computing offers three types of services: platform as a service (PAAS), infrastructure as a service (IAAS), and software as a service (SAAS)[1,2,3]. Cloud computing services make use of emerging computing technologies such as fog computing and edge computing. Cloud computing's centralized management is a significant impediment to resource allocation and task scheduling. The management of task scheduling and resource allocation handles load balancing. The unbalanced scenario of resources and allocation management suffers from an unsupported capacity of load and a decline in the system capacity of cloud computing[4,5,6]. Load balancing approaches deal with the conflict between resource allocation and task scheduling. The load balancing approach is divided into two sections: static load balancing and dynamic load balancing. The static load balancing approach uses conventional scheduling algorithms such as FCFS, SJF, and round robin (RR). For resource and task allocation, the dynamic load balancing approach employs a partition-based scheduling algorithm and a heuristic-based function. Because of the exponential growth in the number of customers using the service, an effective load balancing scheme is essential in cloud computing. It is important in digital devices, servers, and networks because it allows for maximum resource utilization, increased scalability, and the elimination of disruptions and over-supply. In a cloud, there are various types of loads such as memory, network, and so on. Virtual machine is core component of cloud computing. The utilization and processing of virtual machine depends the performance of cloud computing in manners of make span time and execution time[7,8,9]. The several authors proposed various methods regrading processing of virtual machine. The inter-relation of virtual machine and coupling of virtual machine improves the performance of cloud computing. Swarm intelligence is now being used to solve many real-time problems such as optimization, robot manipulation, and routing. Swarm Intelligence (SI) is a type of artificial brain self-discipline (AI) that is stimulated by the collective behavior of ants, cock of birds, schooling fish, bees, termites, and worms. Swarm brain method is now used in many fields such as robustness, mobile telecommunication networks, and so on. Ant colony optimization (ACO), for example, is a high-quality swarm Genius optimization strategy. It is difficult to solve NP difficult problems using traditional search methods, so swarm talent strategies come into play. As recent trends, eminent students used particle swarm brain algorithm, firefly algorithm, and genetic algorithm to aid scheduling in cloud computing, which results in higher answer [10] and synthetic bee colony (ABC). Cuckoo-search, the bat algorithm, the firefly algorithm (FA), krill herd bio-inspired optimization algorithm, and clustering algorithms are some of the more promising swarm - Genius optimization methods available today. Metaheuristic algorithms include swarm intelligence-based algorithms. One of the most important swarm intelligence-based algorithms is the firefly algorithm. Several changes have occurred in distributed computing environments. Cluster emergence is the first change that connects a group of loosely or tightly coupled systems to work together. Then, grid computing is created to address the issue of cluster systems only being able to use local resources. As a result, the grid aggregates all available heterogeneous resources from geographically distributed systems. The remaining of paper describes as in section II related work, in section III proposed methodology, in section IV describes experimental analysis of proposed algorithm and finally conclude in section V.

II. RELATED WORK

The distributed computing offers several services over internet based on service request. Recently several authors proposed swarm intelligence and machine learning based algorithm for task scheduling and load balancing. The recently contributed work describes here. In [1], the QMTSF for cloud task scheduling was proposed, introducing the UQRL algorithm for task assignment within servers. The superiority of QMTSF over PSO, random, and RR scheduling was validated. Traditional algorithms are static, limiting adaptability and efficiency. The UQRL algorithm may deviate from local optimal solutions. In [2], a machine learning model to enhance virtual machine migration was proposed, presenting a solution-based approach to improve VM migration in cloud computing. No explicit limitations were mentioned in the paper. In [3], a cloud platform to enhance English teaching with personalized content and solutions was developed. The platform employs machine learning for personalized education and conducts experiments on online English teaching, proposing constructive strategies. However, data loss due to regression analysis for anomaly detection and a linear increase in system delay time with increasing concurrent operators were noted. In [4], the focus was on improving the efficiency of the GNB ML algorithm for DDOS detection in the cloud. GNB accuracy was enhanced through feature selection and data pre-processing. Limitations include the zero-frequency issue and the assumption of feature independence, resulting in lower accuracy compared to other classifiers. In [5], a novel approach for early intrusion detection in cloud computing using time series data was presented. A collaborative feature selection model was integrated with anomaly detection techniques. Concerns about high-dimensional data prediction accuracy and challenges in cross-validation for time series evaluation were addressed. In [6], new feature-driven content for email phishing detection using SVM, NB, and LSTM classifiers was explored. There is limited discussion on dataset pre-processing and feature selection. In [7], an energy-efficient VM allocation and migration algorithm using SESA was proposed, incorporating cosine similarity and bandwidth utilization for improved performance. An artificial bee colony was employed for destination node selection in migration, modifying state-of-the-art techniques for better power efficiency and utility. Noted limitations include the lack of real-world implementation and limited comparison with other state-of-the-art techniques. In [8], anomaly detection using an RF classifier and feature engineering to enhance IDS in the cloud was proposed. Despite technological advancements, cloud providers face security limits. The IDS model evaluation focused on ACC, recall, and precision metrics. In [9], an RL model to minimize energy consumption in cloud data centers was proposed, addressing the inefficiency of existing models in minimizing resource energy. The unsuitability of single-agent RL models for the entire cloud system was highlighted. In [10], ML techniques for cloud energy efficiency were reviewed, highlighting the use of deep reinforcement learning, TensorFlow, and CloudSim. Poor model accuracy, optimization issues, resource allocation problems, high computation requirements, and centralized approaches were identified as limitations. In [11], cloud computing performance was enhanced using machine learning methods, validating a new algorithm to mitigate cloud security risks. Limitations include latency, data streaming, and big data management challenges. The proposed system requires more extensive testing with larger data sets. In [12], load balancing, resource distribution, and power management were integrated using the WOA algorithm. Task validation, reaction period, and power expenditure in cloud systems were optimized. The lack of task deployment policy consideration and the need for multi-objective optimization in LB-RC were highlighted. Hybrid, firefly, and IPSO techniques were compared, noting reduced energy efficiency. TSFPA improved task duration but lacked a detailed comparison. In [13], the impact of bug classification on cloud system performance and efficiency was discussed. Models can become outdated due to evolving cloud computing applications, requiring constant updates and retraining to maintain accuracy. In [14], mobile cloud computing for emergency scenarios was demonstrated with an efficient simulation model. Cloud nodes offered affordable assistance, but higher restrictions degraded network performance, and inaccurate decision-making occurred due to weight constraints. In [15], HHRL for task scheduling in cloud computing was proposed, introducing a task complexity estimation method based on linear regression. Experiments with CloudSim and a real cloud server for validation were conducted, noting challenges in computational complexity estimation and accurately estimating task complexity in real application scenarios. In [16], network security in IoT cloud computing using machine learning techniques was explored, providing insights into benefits, challenges, and considerations. Predictive maintenance and proactive security measures were addressed, noting challenges in machine learning adoption for network security, the importance of data pre-processing, and the unsuitability of the fragmentation scheme for large datasets. In [17], the HealthEdge smart healthcare framework for type 2 diabetes prediction was proposed, evaluating the system using two common machine learning algorithms on real-life datasets. The gap was addressed by proposing an integrated IoT-edge cloud computing system, but limited diversity in datasets and the inefficiency of the SMOTE algorithm for producing better training due to oversampling were noted. In [18], cloud security was enhanced through machine learning for threat identification and response, formulating a practical strategy for ML predictions in industrial cloud cybersecurity. The lack of a security framework in the data classification model and the failure to categorize some attacks in anomaly detection and classification were identified. In [19], the cloud-to-edge ecosystem with ML applications was reviewed, identifying research challenges and future directions for computational resources. ML algorithms for cloud-to-edge resource management were recommended, noting the lack of a comprehensive review on cloud-to-edge computational resource management and the need for research on ML approaches to address resource management challenges. In [20], a solution to task scheduling with an advanced ordinal optimization technique was offered, presenting a mathematical equation for scheduling tasks with the shortest makespan period. The challenge of large search spaces in optimal task scheduling was addressed, aiming to reduce scheduling overhead effectively. In [21], the HMHHO algorithm for task scheduling in mobile cloud computing was proposed, integrating a local search algorithm to enhance performance. Runtime performance as a drawback in various test data and the algorithm's sensitivity to parameter values affecting optimal performance were noted. In [22], the Coot Optimization Algorithm for task scheduling was enhanced, improving the density-based clustering method for task classification. Scheduling efficiency deterioration due to various parameters affecting cloud system performance and challenges including security, cost management, multi-cloud disparities, and interoperability issues were identified. In [23], the HWACOA algorithm for task scheduling in cloud computing was introduced, comparing it with existing algorithms using experimental simulation results. Task scheduling performance and cloud operations were aimed to be enhanced, noting drawbacks in existing ACO algorithms and the theoretical superiority of HWACO due to the weight concept. In [24], the FTTATS algorithm for fault-tolerant trust-based task scheduling in cloud computing was proposed, utilizing Harris Hawks Optimization for effective task scheduling. FTTATS was evaluated against ACO, PSO, and GA algorithms, noting significant improvements without explicit mention of limitations. In [25], whale optimization was used to minimize SLA violations in cloud computing, proposing a task-scheduling mechanism to improve makespan and SLA violations. The failure of existing algorithms in highly dynamic cloud environments and challenges with evolving data and computing needs in on-premises infrastructure were noted.

III. PROPOSED METHODOLOGY

The proposed algorithm accelerates task scheduling in cloud computing by combining Particle Swarm Optimization (PSO) and Firefly (FF) algorithms. The FF algorithm couples with virtual machines, while PSO allocates resources for job execution[15,16,17]. Together, FF and PSO provide global and local optimal solutions for cloud task scheduling. The hybrid algorithm processes different tasks, denoted as t1, t2, ..., tn. In this context, W represents the weight factor for all tasks assigned within the cloud environment. M indicates the number of firefly mates, v1 and v2 denote the velocity of particle agents, and c1 and c2 are constants in the PSO algorithm. The allocation process of tasks is described below. Begin 1 define the value of task set T={ t1,t2,…………..,tn} . the VMs comprising modules of coupling computing machinery installed upon cloud computers disposed within a data center, the cloud computing environment further comprising a cloud operating system and a data center administration server operably coupled to the VMs, the method comprising. deploying, by the cloud operating system, an instance of a VM, including flagging the instance of a VM for autonomic scaling and executing a data processing workload on the instance of a VM.

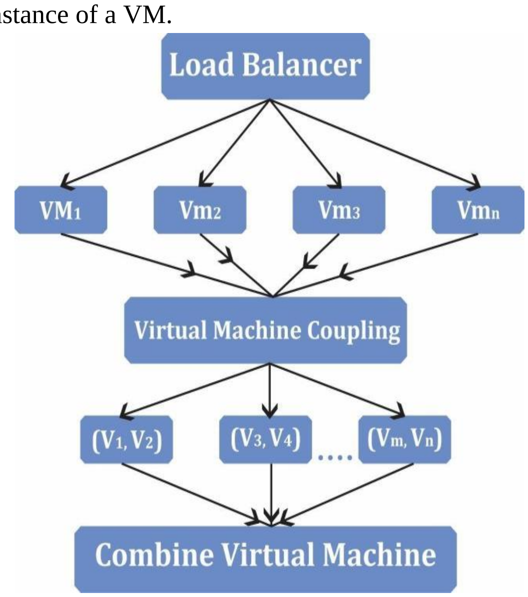

Figure1 show that process of coupling of virtual machine

IV. EXPERIMENTAL ANALYSIS

To verify the utility of the proposed algorithm for task scheduling in cloudSim Simulators. The CloudSim simulator is an open-source toolkit for simulating cloud computing environments. It has been developed in the cloud lab of Melbourne university. This simulator contains a data center, virtual machine, traffic load, users and scheduling algorithm. The complete simulation process deal with the java programing language. The proposed algorithm is implemented in the scheduling algorithm with an existing algorithm for the simulation of the dedicated scenario for different groups of loads and measuring standard parameters of the simulation [24,25,25].

Response Time

The response time of a task refers to the time intervals among tasks to arrive into the system until its completion. Response time Re is expressed as

Re = Tc − Ta + Tt……………………(1)

where Tc represents the time required to complete a task, Ta represents the arrival time of a task, and Tt represents the transfer time of a task.

Make-span

Make-span is defined as the total time taken to process a set of tasks for its complete execution. Make-span Ma is represented as

Ma = max (CT)……………………(2)

where max (CT) is the maximum time required to complete all tasks.

Resource Utilization

Resource utilization denotes the number of resources required during task execution. Resource utilization Ru is expressed as

$$Ru = \frac{T_c}{M_a X\ N} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\ (3)$$

where Tc represents the time taken to complete a task, Ma represents make-span and N represents number of resources.

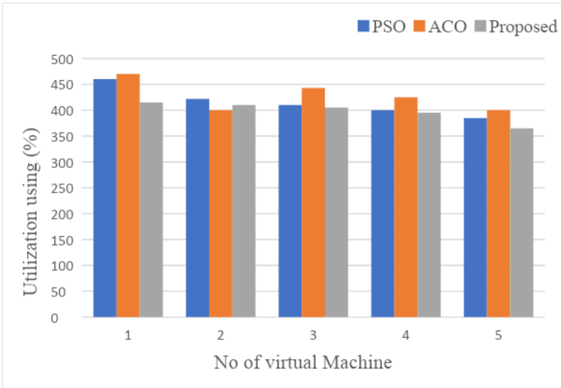

Figure 2: Performance evaluation of the number of virtual machines and VM using (%) For the PSO, ACO, and Proposed methods,

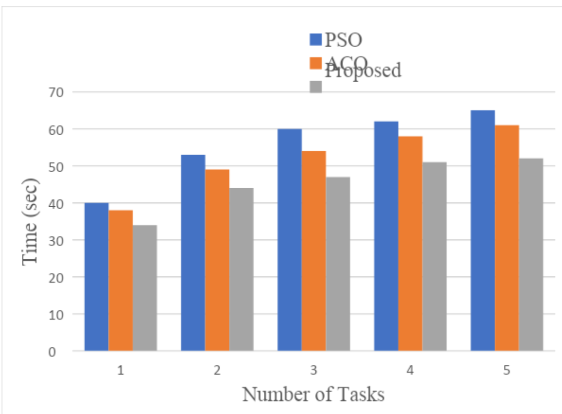

Figure 3: Performance analysis Number of tasks and time (sec) for using methods PSO, ACO, and Proposed.

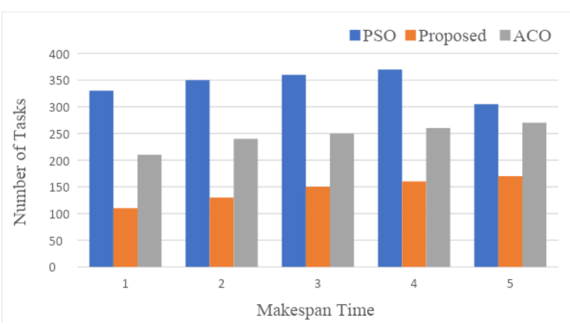

Figure 4: Method for analysing the performance of the number of tasks versus make span time PSO, proposed, ACO,

V. CONCLUSION & FUTURE WORK

The proposed optimization algorithm integrates virtual machines within the constraints of the firefly algorithm, minimizing task execution time in data centers. In this hybrid technique, firefly optimization narrows the search for the optimal solution, while particle swarm optimization identifies the best answer. This strategy focuses on predictive workload allocation, incorporating resource scalability and a load-balancing model to facilitate workflow execution and maximize the use of evenly distributed VMs. Experimental results indicate that the suggested hybrid load-balancing method outperformed existing systems in terms of time duration, resource utilization, imbalance degree, and task migrations. The findings demonstrate that the proposed system offers superior response time and higher performance compared to prevailing techniques. The comparison reveals that the proposed algorithm achieves 49.23% shorter execution time than the existing ACO algorithm, 60.5% lower communication cost than the PSO algorithm, and 43.9% higher resource utilization than the existing ACO algorithm.

REFERENCES

[1] Wang, Yugui, Shizhong Dong, and Weibei Fan. "Task scheduling mechanism based on reinforcement learning in cloud computing." Mathematics 11, no. 15 (2023): 3364.

[2] Belgacem, Ali, Saïd Mahmoudi, and Mohamed Amine Ferrag. "A machine learning model for improving virtual machine migration in cloud computing." The Journal of Supercomputing 79, no. 9 (2023): 9486-9508.

[3] Zhang, Peili. "Cloud computing English teaching application platform based on machine learning algorithm." Soft Computing (2023): 1-13.

[4] Naiem, Sarah, Ayman E. Khedr, Mohamed Marie, and Amira M. Idrees. "Enhancing the efficiency of gaussian naïve bayes machine learning classifier in the detection of ddos in cloud computing." IEEE Access (2023).

[5] Al-Ghuwairi, Abdel-Rahman, Yousef Sharrab, Dimah Al-Fraihat, Majed AlElaimat, Ayoub Alsarhan, and Abdulmohsen Algarni. "Intrusion detection in cloud computing based on time series anomalies utilizing machine learning." Journal of Cloud Computing 12, no. 1 (2023): 127.

[6] Butt, Umer Ahmed, Rashid Amin, Hamza Aldabbas, Senthilkumar Mohan, Bader Alouffi, and Ali Ahmadian. "Cloud-based email phishing attack using machine and deep learning algorithm." Complex & Intelligent Systems 9, no. 3 (2023): 3043-3070.

[7] Kaur, Amandeep, Saurabh Kumar, Deepali Gupta, Yasir Hamid, Monia Hamdi, Amel Ksibi, Hela Elmannai, and Shilpa Saini. "Algorithmic approach to virtual machine migration in cloud computing with updated SESA algorithm." Sensors 23, no. 13 (2023): 6117.

[8] Attou, Hanaa, Azidine Guezzaz, Said Benkirane, Mourade Azrour, and Yousef Farhaoui. "Cloud-based intrusion detection approach using machine learning techniques." Big Data Mining and Analytics 6, no. 3 (2023): 311-320.

[9] Prabha, B., J. Thangakumar, and K. Ramesh. "Reinforcement learning based energy consolidation model for efficient cloud computing system." Appl. Math. Inf. Sci 17, no. 1 (2023): 67-77.

[10] Puso, Nomsa, Tshiamo Sigwele, and Oba Zubair Mustapha. "Machine Learning Centered Energy Optimization In Cloud Computing: A Review." Indonesian Journal of Electrical Engineering and Informatics (IJEEI) 11, no. 3 (2023): 834-853.

[11] Sultan, Nawar. "Study on the Design of Algorithm Based on Machine Learning to Improve Cloud Computing." (2023).

[12] Bhuiyan, Md Shamsuzzaman, Amatur Rahman Sarah, Shakib Khan, Al Kawsar, and Ahmed Wasif Reza. "An Improved Framework for Power Efficiency and Resource Distribution in Cloud Computing Using Machine Learning Algorithm." In International Conference on Big Data, IoT and Machine Learning, pp. 685-697. Singapore: Springer Nature Singapore, 2023.

[13] Tabassum, Nadia, Abdallah Namoun, Tahir Alyas, Ali Tufail, Muhammad Taqi, and Ki-Hyung Kim. "Classification of bugs in cloud computing applications using machine learning techniques." Applied Sciences 13, no. 5 (2023): 2880.

[14] Hai, Tao, Jincheng Zhou, Ye Lu, Dayang NA Jawawi, Dan Wang, Shitharth Selvarajan, Hariprasath Manoharan, and Ebuka Ibeke. "An archetypal determination of mobile cloud computing for emergency applications using decision tree algorithm." Journal of cloud computing 12, no. 1 (2023): 73.

[15] Yin, Lei, Chang Sun, Ming Gao, Yadong Fang, Ming Li, and Fengyu Zhou. "Hyper-Heuristic Task Scheduling Algorithm Based on Reinforcement Learning in Cloud Computing." Intelligent Automation & Soft Computing 37, no. 2 (2023).

[16] Naeem, Humaira. "Analysis of Network Security in IoT-based Cloud Computing Using Machine Learning." International Journal for Electronic Crime Investigation 7, no. 2 (2023).

[17] Hennebelle, Alain, Huned Materwala, and Leila Ismail. "HealthEdge: a machine learning-based smart healthcare framework for prediction of type 2 diabetes in an integrated IoT, edge, and cloud computing system." Procedia Computer Science 220 (2023): 331-338.

[18] Abbas, Zaheer, and Seunghwan Myeong. "Enhancing industrial cyber security, focusing on formulating a practical strategy for making predictions through machine learning tools in cloud computing environment." Electronics 12, no. 12 (2023): 2650.

[19] Adeniyi, Emmanuel A., Sunday Adeola Ajagbe, Olukayode A. Oki, Aminat Omotayo Adebayo, and Oyebola Olasupo. "Application of Machine Learning Algorithm in Cloud-to-edge Computing: Analysis and Limitations." 2023 IEEE AFRICON (2023): 1-6.

[20] Yadav, Monika, and Atul Mishra. "An enhanced ordinal optimization with lower scheduling overhead based novel approach for task scheduling in cloud computing environment." Journal of Cloud Computing 12, no. 1 (2023): 8.

[21] Saemi, Behzad, Ali Asghar Rahmani Hosseinbadi, Azadeh Khodadadi, SeyedSaeid Mirkamali, and Ajith Abraham. "Solving task scheduling problem in mobile cloud computing using the hybrid multi-objective Harris Hawks optimization algorithm." IEEE Access (2023).

[22] Karimunnisa, Syed, and Yellamma Pachipala. "Task Classification and Scheduling Using

Enhanced Coot Optimization in Cloud Computing." International Journal of Intelligent Engineering & Systems 16, no. 5 (2023).

[23] Chandrashekar, Chirag, Pradeep Krishnadoss, Vijayakumar Kedalu Poornachary, Balasundaram Ananthakrishnan, and Kumar Rangasamy. "HWACOA scheduler: Hybrid weighted ant colony optimization algorithm for task scheduling in cloud computing." Applied Sciences 13, no. 6 (2023): 3433.

[24] Mangalampalli, Sudheer, Ganesh Reddy Karri, Amit Gupta, Tulika Chakrabarti, Sri Hari Nallamala, Prasun Chakrabarti, Bhuvan Unhelkar, and Martin Margala. "Fault-Tolerant Trust-Based Task Scheduling Algorithm Using Harris Hawks Optimization in Cloud Computing." Sensors 23, no. 18 (2023): 8009.

[25] Mangalampalli, Sudheer, Sangram Keshari Swain, Ganesh Reddy Karri, and Satyasis Mishra. "SLA Aware Task-Scheduling Algorithm in Cloud Computing Using Whale Optimization Algorithm." Scientific Programming 2023, no. 1 (2023): 8830895.