The Scientific Temper (2025) Vol. 16 (5): 4270-4283
E-ISSN: 2231-6396, ISSN: 0976-8653
Doi: 10.58414/SCIENTIFICTEMPER.2025.16.5.12
https://scientifictemper.com/

RESEARCH ARTICLE

A hybrid approach using attention bidirectional gated recurrent unit and weight-adaptive sparrow search optimization for cloud load balancing

V. Infine Sinduja[1*], P. Joesph Charles[2]

Abstract

With the evolution of cloud computing (CC) technologies, there is a growing insistence on the maximum utilization of cloud resources, thereby increasing the computing power consumption of cloud systems. Cloud's virtual machines (VMs) consolidation imparts a practical mechanism to minimize the energy consumption of cloud data centers (DC). Efficient consolidation and migration of VM in the absence of infringing Service Level Agreement (SLA) can be arrived at by making decisions proactively based on the cloud's future workload prediction. Efficient load balancing, another major issue of CC also depends on accurate forecasting of resource usage. Cloud workload traces reveal both periodic and non-periodic patterns with the unexpected peak of load. As a result, it is very demanding for the prediction models to accurately anticipate future workload. This prompted us to propose a method called, Attention Bidirectional Gated and Weight-adaptive Sparrow Search Optimization (ABiG-WSSO) to accurately forecast future workload with minimal makespan and overhead. The proposed ABiG-WSSO method includes Attention Bidirectional Gated Recurrent Unit (ABiGRU) and Weight-adaptive Sparrow Search Optimization (WSSO). Attention Bidirectional Gated Recurrent Unit (ABiGRU) is initially designed that along with the use of Bidirectional Gated Recurrent Unit (BiGRU) and adaptation of attention mechanism aids in predicting future cloud load requirements accurately. Next, Weight-adaptive Sparrow Search Optimization (WSSO) algorithm is employed in fine-tuning the parameters of the ABiGRU model for accurate and optimal load balancing performance. The WSSO algorithm is applied to optimize ABiGRU model hyperparameters (i.e. learning rate), to enhance its prediction accuracy. Comprehensive simulations are carried out using the gwa-bitbrains dataset to verify the efficiency of the proposed ABiG-WSSO method in boosting the distribution of resources and cloud load balancing. The proposed method achieves comparatively better results in terms of better makespan time, energy consumption, associated overhead and throughput.

Keywords: Cloud Computing, Service Level Agreement, Attention, Bidirectional Gated Recurrent Unit, Weight-adaptive, Sparrow Search

[1]Department of Computer St. Joseph's College (Autonomous) Affiliated to Bharathidasan University, Thiruchirappalli, India.

[2]Department of Computer Science, St. Joseph's College (Autonomous) (Affiliated to Bharathidasan University), Thiruchirappalli, India.

***Corresponding Author:** V. Infine Sinduja, Department of Computer St. Joseph's College (Autonomous) Affiliated to Bharathidasan University, Thiruchirappalli, India., E-Mail: infinesinduja@gmail.com

How to cite this article: Sinduja, V.I., Charles, P.J. (2025). A hybrid approach using attention bidirectional gated recurrent unit and weight-adaptive sparrow search optimization for cloud load balancing. The Scientific Temper, **16**(5):4270-4283.

Doi: 10.58414/SCIENTIFICTEMPER.2025.16.5.12

Source of support: Nil

Conflict of interest: None.

Introduction

Cloud Computing (CC) framework plays a crucial part as far as sophisticated business operations, imparting a wide range of services to name a few being software accessible via web browsers and platforms for designing cloud-based applications. CC framework makes certain the optimal usage of computing resources utilizing its dynamic service model. Moreover, CC services necessitate flexible distribution of computing resources with arbitrary resource scalability that assists in implementing a Quality-of-Service (QoS) with the paramount minor resource expenses. Nevertheless, in complicated CC framework with varying workloads, it might be demanding to perform arbitrary distribution of resource for heterogeneous applications.

Effective allocation of task and balancing of load are demanding to circumvent both under loading or overloading structures that can bring about execution delays or machine failures. An Enhanced Dynamic Load

Received: 04/04/2025 **Accepted:** 07/05/2025 **Published**: 31/05/2025

Balancing (EDLB) method was proposed in (Zhanuzak, R., Ala'anzy, M. A., Othman, M., & Algarni, A., 2024) with the intent of scheduling task in an optimal fashion and allocation resource in CC environments. Upon comparison to standard methods that depend on static selection of Virtual Machine (VM), the EDLB identified optimal cloudlet placement arbitrarily in real-time.

The EDLB method allocated cloudlets to VMs proactively on the basis of current system states and Service Level Agreement (SLA) deadlines. Moreover, if the VM did not converged the cloudlet deadline the method redirected to secondary data in order to make certain optimal allocation with minimal makespan, execution time and maximal resource utilization. Additionally to provide deeper insights into its effectiveness by including a broader range of QoS parameters, such as energy consumption and overhead, in this work an Attention Bidirectional Gated Recurrent Unit (ABiGRU) is proposed in the ABiG-WSSO method.

The major issue with VM integration methods is the trade-off between cost efficiency, performance, quality of service, optimal utilization of resources and so on. Nevertheless, there remains a major issue with obtaining variations in workload owing to the constrained provisioning of resource level.

A hybrid method utilizing deep learning with Particle Swarm Optimization and Genetic Algorithm (DPSO-GA) was proposed (Simaiya, S., Lilhore, U. K., Sharma, Y. K., Brahma Rao, K. B. V., Maheswara Rao, V. V. R., Baliyan, A., Bijalwan, A., & Alroobaea, R., 2024) for making certain arbitrary workload in CC environment via two phases. The first phase employed PSO-GA model addressed the issues related to prediction by integrating the advantages of these two models methods in fine-tuning Hyperparameters. The second phase employed CNN-LSTM for extracting complex discriminating features acquired from VM workload statistic with improved precision, recall and accuracy factors. Additionally to provide deeper insights into the key metric for monitoring performance and identifying potential bottlenecks, in this work throughput involved in cloud load balancing will be addressed using Weight-adaptive Sparrow Search Optimization model in the ABiG-WSSO method.

A key issue remained in network speed optimization while preserving fairness. Several monitoring solutions have been designed to minimize these load imbalances. To accomplish the cloud load balancing, this an intelligent virtual machine programming called, an Improved Lion Optimization (ILO) with Min-Max Algorithm was proposed (Adaikalaraj, J. R., & Chandrasekar, C., 2023) for parallelization, therefore ensuring load balancing. One of the major areas as far as business and computing technologies are concerned, high performance computing steal the major portion to meet business continuity and real-time needs. Nevertheless, several business and technology organizations are in the process of enhancing high performance to make certain the availability of the system at all times.

(Kamila, N. K., Frnda, J., Pani, S. K., Das, R., Islam, S. M. N., Bharti, P. K., & Muduli, K., 2022) the concept of high-performance computing was integrated with artificial intelligence machine learning in cloud platforms, therefore resulting in both load balancing and cost savings considerably. The cloud computing environment has been allocated some load that can be either overloaded underloaded or balanced subject to the cloud architecture design and cloud user requested tasks. A predominant element of task scheduling in CC environment is the balancing of workloads according to the dependency or independency nature of VMs. To address on these limitations, a novel Load Balancing of Virtual Machine (LBVM) in CC was proposed (Muneeswari, G., Madavarapu, J. B., Ramani, R., Rajeshkumar, C., & Singh, C. J. C., 2024) with improved detection rate and accuracy.

Despite cloud computing give prominence to attractive advantages, certain unpredictable circumstances, e.g., heavy workload can result in inefficient allocation of resources. To address these issues and control the drawbacks, an efficient deep neural network employing supervised learning for cloud workload prediction was proposed in (Xu, M., Song, C., Wu, H., Gill, S. S., Ye, K., & Xu, C., 2022). With this design accurate and efficient cloud workload prediction was made with minimal mean square error. Yet another hybrid optimization method was designed (Saini, H., Singh, G., Kaur, A., Saini, S., Wani, N. A., Chopra, V., Akhtar, Z., & Bhat, S. A., 2024) to ensure resource efficiency.

Owing to the emergence of large number of incoming cloud user requested tasks in CC environment, load balancing is considered as a paramount issue. (Pradhan, A., Bisoy, S. K., Kautish, S., Jasser, M. B., & Mohamed, A. W., 2022) a hybrid method combining Deep Reinforcement Learning and parallel designing of Particle Swarm Optimization with the intent of addressing the load balancing issue with greater accuracy and high speed was proposed. (Lohumi, Y., Gangodkar, D., Srivasatava, P., Khan, M. Z., Alahmadi, A., & Alahmadi, A. H., 2023) A state of the art load balancing techniques was investigated.

Enormous scalability is feasible due to load balancing. Owing to the evolution of the cloud, large numbers of service provisioning requests are said to be brought about from cloud users. (Mathanraj, E., & Reddy, R. N., 2024) a load balancing technique using principal component gradient was proposed with the objective of balancing workload in a cloud server for controlling huge workload within a stipulated time period. This load balancing technique achieved higher load balance with minimal execution time. Neural networks were employed (Yildirim, E., & Akon, A., 2023) to improve overall throughput rate. For non-stationary data patterns hybridization of gated recurrent unit and

convolutional neural network was designed (Han, H., Neira-Molina, H., Khan, A., Fang, M., Mahmoud, H. A., Awwad, E. M., Ahmed, B., & Ghadi, Y. Y., 2024) for robust and accurate time series forecasting.

For the issues of low accuracy and low efficiency of most load forecasting methods, a load forecasting method employing enhanced deep learning in CC environment was proposed (Zhang, K., Guo, W., Feng, J., & Liu, M., 2021). For load classification deep belief network was used that in turn ensured minimal mean prediction error. Yet another bio inspired algorithm employing genetic algorithms with ant colony was designed (Brahmam, M. G., & Anand, V. R., 2024), therefore minimizing energy consumption and ensuring improved load balancing. A load balancing mechanism employing deep reinforcement technique was presented (Lahande, P. V., Kaveri, P. R., Saini, J. R., Kotecha, K., & Alfarhood, S., 2023) with the objective of providing best quality of service.

Problem statement

- Load balancing in Cloud Computing environment is a major user experience concern, with existing resource optimization method often- falling short in throughput and energy consumption for better application performance.
- Traditional load balancing methods face challenges in accurately handling complicated temporal and accurately anticipating future workload in distributed datacenter datasets.
- The problem addressed is the requirement for an energy and throughput efficient future workload balancing by fine hyperparameters or learning rate for better application performance and cloud user experience.

Contribution of the study

An efficient combination of several methods can give rise to hybrid methods. In this paper, a novel hybrid method called, Attention Bidirectional Gated and Weight-adaptive Sparrow Search Optimization (ABiG-WSSO) is presented to accurately forecast future workload with minimal energy consumption and makespan. The major contribution of the ABiG-WSSO method is stated below:

- To propose a hybrid method combining Attention Bidirectional Gated and Weight-adaptive Sparrow Search Optimization (ABiG-WSSO) to improve overall throughput and makespan involved in cloud load balancing.
- To enhance predicting future cloud load requirement accurately, with the intent of facilitating less energy consumption and overall overhead.
- Captures complicated and temporal patterns by fine-tuning the parameters of ABiGRU, therefore ensuring improved throughput with minimal makespan in generating optimal load balancing performance.
- The efficiency of the proposed ABiG-WSSO method has been determined using evaluation metrics such as makespan, energy consumption, throughput and overhead etc.

Article organization

The rest of this article is organized as follows. Section 2 discusses the related work for load balancing prediction in cloud computing environments. The proposed method based on deep learning is introduced in Section 3 including the framework, pseudo code representation with the aid of figurative representations. Section 4 introduces the details of our experiments that apply dataset obtained from realistic traces to predict workloads and exhibit the feasibility of our method to boost load balancing. Moreover the performance metrics and discussion with the aid of graph and table representations are also provided with detailed comparison to existing related work. Finally, conclusions are given in Section 5 by summarizing its concept and content.

Related Works

Integration methods on virtual machine (VM) have efficiently manifested an optimized load balancing as far as CC environment is concerned. The main issue with VM integration methods is the trade-off between quality of service, optimal utilization of resource and cost efficiency with service level agreement violations. Deep Learning algorithms are found to be of extensively utilized on cloud load balancing. Nevertheless, there still remains issue with obtaining workload owing to the constrained nature of resource-level provisioning.

A holistic review of load balancing was investigated (Devi, N., Dalal, S., Solanki, K., Dalal, S., Lilhore, U. K., Simaiya, S., & Nuris, N., 2024). Effective allocation of resource and distribution of workload are paramount to make certain continuous and reliable service are said to be focused. However, with increase in the data volumes a novel optimization method employing deep learning to handle these issues was proposed (Lilhore, U. K., Simaiya, S., Sharma, Y. K., Rai, A. K., Padmaja, S. M., Nabilal, K. V., Kumar, V., Alroobaea, R., & Alsufyani, H., 2025). With this novel optimization method reduced energy consumption and improved throughput considerably. A deep learning algorithm called Particle Swarm Intelligence and Genetic Algorithm was presented (Simaiya, S., Lilhore, U. K., Sharma, Y. K., Brahma Rao, K. B. V., Maheswara Rao, V. V. R., Baliyan, A., Bijalwan, A., & Alroobaea, R., 2024) for dynamic workload provisioning in CC environment. To provide the best quality of service a reinforcement learning technique was proposed (Lahande, P. V., Kaveri, P. R., Saini, J. R., Kotecha, K., & Alfarhood, S., 2023). A systematic literature review on load balancing and scheduling of task in CC was investigated (Devi, N., Dalal, S., Solanki, K., Dalal, S., Lilhore, U. K., Simaiya, S., & Nuris, N., 2024).

To harvest the advantages of several cloud services, cloud industries should adopt holistic resources scheduling mechanisms. By deploying effective deep learning techniques, several cloud traffics' potential issues can be solved. (Ikhlasse, H., Benjamin, D., Vincent, C., & Medromi, H., 2022) a Bidirectional Gated Recurrent Unit based on Stacked Denoising Autoencoders to forecast simultaneously future hourly virtual CPU, memory, and storage utilizations was proposed. Yet another hybrid offloading method was presented in (Sulimani, H., Sulimani, R., Ramezani, F., Naderpour, M., Huo, H., Jan, T., & Prasad, M., 2024) to improve load balancing and enhance overall system performance.

Effective allocation of resource and distribution of workload are paramount to making certain the continuous and reliable service with increasing data volumes is focused. (Lilhore, U. K., Simaiya, S., Sharma, Y. K., Rai, A. K., Padmaja, S. M., Nabilal, K. V., Kumar, V., Alroobaea, R., & Alsufyani, H., 2025) Deep Q-Networks (DQN) was integrated with Proximal Policy Optimization (PPO) to focus on the arbitrary features of IoT applications, therefore improving energy consumption and task scheduling time considerably. Considerable literature review suggests that the prospective of recurrent neural networks with attention mechanisms is not adequately investigated and applied to CC. To address this gap, recurrent neural networks with and without attention layers for load forecasting was proposed (Predić, B., Jovanovic, L., Simic, V., Bacanin, N., Zivkovic, M., Spalevic, P., Budimirovic, N., & Dobrojevic, M., 2023), therefore learning to minimal mean square error.

In spite of cloud offloading issues have been extensively researched under several backgrounds and methodologies, load balance, which is a paramount method in CC environment's are considered to make certain the full equipment of resources is being used, has not yet been accounted for. To fill this issue, a dynamic load balance-aware offloading technique employing optimal control numerical methods was presented (Fan, Y., 2024) with improved accuracy. However the average waiting time was not concentrated and to focus on this aspect, an enhanced round robin technique was designed (Zohora, M. F., Farhin, F., & Kaiser, M. S., 2024). By using this enhanced technique resulted both the in improvement of average waiting time and ensured optimality. Yet another hybrid evolutionary machine learning technique was presented (Sharma, A., Rani, S., & Driss, M., 2024) to effectively handle data streams in real time. A survey of machine learning techniques for balancing load in CC environment was designed (Gures, E., Shayea, I., Ergen, M., Azmi, M. H., & El-Saleh, A. A., 2022). Yet another comprehensive overview of load balancing methods to make certain efficient communication processes was carried out and investigated (Farahi, R., 2025). A comprehensive study of distinct load balancing architectures in CC was

Table 1: Advantages and disadvantages of each load balancing algorithm /technique

Reference number	*Techniques/ Algorithm*	*Advantages*	*Disadvantages*
Zhanuzak, R, et al., 2024	Enhanced Dynamic Load Balancing (EDLB) method	make certain optimal allocation with minimal makespan, execution time and maximal resource utilization	Did not included energy consumption and overhead
Simaiya, S, et al., 2024	Particle Swarm Optimization and Genetic Algorithm	Extract discriminating features from VM workload with improved precision, recall and accuracy	Throughput involved in cloud load balancing
Adaikalaraj, J. R, et al., 2023	Improved lion optimization with Min-Max algorithm	Minimized energy consumption	Lack makespan analysis
Muneeswari, G, et al., 2023	Bi-LSTM	Minimized migration time	Accuracy
Saini, H., et al., 2024	Time-aware modified best fit decreasing (T-MBFD) algorithm	Improved sensitivity and specificity	Did not focus energy consumption
Pradhan, A, et al., 2022	Deep Reinforcement Learning with Parallel Particle Swarm Optimization (DRLPPSO)	Task scheduling efficiency	Did not focus on overhead
Mathanraj, E, et al., 2024	principal component gradient round robin load balancing (PCGRLB) technique	Balance workload accurately	Did not concentrate on makespan time
Yildirim, E, et al., 2023	Neural networks	Minimized error rate	Lack energy consumption
Han, H, et al., 2024	gated recurrent unit and graph convolutional neural network	Improved predictive accuracy	Did not focus on overhead and energy consumption
Zhang, K, et al., 2021	Improved deep learning	Reduced mean prediction error	Lack precision and accuracy

designed (Narsipuram, M., Rai, A., & Tiwari, A., 2024).

From these literature studies, the author's utilization of CC has made it feasible for line of works to perform resource allocation between several cloud service providers with the intent of managing load. The authors utilized some distinct types of materials and methods to manage the work load or load balancing. The load balancing techniques now in use depend on a number of task parameters. Most methods balance load by employing optimization models. A review of the literature discloses that deep learning-based load balancing, may substantially improve the workload distribution that is balanced. To overcome these limitations a novel ABiG-WSSO method has been proposed in the next section. Comparison between various load balancing methods or algorithms with their advantages and disadvantages is represented in Table 1.

Methodology

Cloud load prediction intents to forecast the future demand for resources like CPU, memory, and storage within a Cloud Computing (CC) environment. Accurate predictions assist cloud service providers and cloud users optimize allocations of resources and ward off congestions. In this section a method called, Attention Bidirectional Gated and Weight-adaptive Sparrow Search Optimization (ABiGRU-WSSO) is proposed. Figure 1 displays the overall methodology of ABiGRU-WSSO method.

As shown in the above figure, the ABiGRU-WSSO method is split into three parts, namely, data collection, future cloud load prediction model and optimal load balancing model via fine-tuning process. Initially in the data collection part, the performance metrics of 1,750 Virtual Machines (VMs) from Bitbrains, a distributed datacenter, is a service provider specialized in managing hosting and business computation for enterprises are obtained.

Next, future cloud load prediction is done by employing Attention Bidirectional Gated Recurrent Unit (ABiGRU) based Cloud Load Prediction model. The ABiGRU in the proposed ABiGRU-WSSO method utilizing Bidirectional Gated Recurrent Unit (BiGRU) and adaptive attention mechanism aims in predicting future cloud load requirements in an energy-efficient manner. Also employing attention mechanism aids in analyzing complicated temporal patterns via forward and backward directions on the most relevant parts of the input data therefore minimizing the overhead considerably. Finally, Weight-adaptive Sparrow Search Optimization (WSSO) algorithm is employed in fine-tuning the parameters of the ABiGRU model for accurate and optimal load balancing performance.

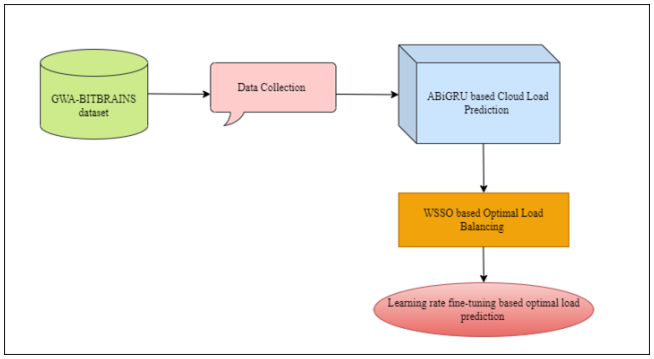

Figure 1: Structure of ABiGRU-WSSO method

Data Collection

The solution of the problem starts by uploading data on the history of the utilization of hardware resources of the data center, the operation of which is be optimized. In the framework of this research, a data set containing a trace of the GWA-T-12 Bitbrains called gwa-bitbrains dataset extracted from https://www.kaggle.com/datasets/gauravdhamane/gwa-bitbrains/data is employed that specializes in providing CC services. The clients in the dataset comprises of large banks, credit card operators and insurance companies. The gwa-bitbrains dataset comprises of a load trace of 1250 virtual machines located in Bitbrains data centers and saved in csv files, the schema of which is provided in Table 2.

With the aid of the information provided in the above schema the design of Attention Bidirectional Gated and Weight-adaptive Sparrow Search Optimization (ABiGRU-WSSO) method is provided in the following sub-sections.

Table 2: Schema of gwa-bitbrains dataset

S. No	*Features*	*Description*	*S. No*	*Features*	*Description*
1	Timestamp	Milliseconds since start of trace	7	Memory usage (%)	Memory actively used
2	CPU cores	Virtual CPU cores provisioned	8	Disk read	Disk read throughput
3	CPU capacity provisioned	Capacity of CPU in terms of MHZ	9	Disk write	Disk write throughput
4	CPU usage (MHZ)	CPU utilization in terms of MHZ	10	Network in	Network received throughput
5	CPU usage (%)	CPU utilization in terms of percentage	11	Network out	Network transmitted throughput
6	Memory provisioned	Memory capacity of VM	12	Day of the week	Number of day of week

Attention Bidirectional Gated Recurrent Unit (ABiGRU) based Cloud Load Prediction

Cloud load predictions encompass of forecasting resource demand of cloud-based application with the intent of optimizing resource allocation and make certain effective functioning. This is achieved via historical data analysis and exploiting deep learning algorithms. In this section, an Attention Bidirectional Gated Recurrent Unit (ABiGRU) based Cloud Load Prediction model is proposed. The ABiGRU based cloud load prediction model is designed along with the utilization of Bidirectional Gated Recurrent Unit (BiGRU) and application of attention mechanism aids in predicting future cloud load requirements accurately.

The bidirectional gated recurrent unit (BiGRU) in the proposed method offers the advantage of efficiently analyzing complicated temporal patterns, both in forward and backward directions permitting for more accurate predictions of future load fluctuations, resulting in enhanced performance. Also, by applying attention mechanism helps in concentrating on relevant factors like, CPU usage and memory usage at each time step and also each cloud server's current load and potential capacity into account for making prediction, therefore enhancing cloud user experience. Figure 2 displays the structure of Attention Bidirectional Gated Recurrent Unit (ABiGRU) based Cloud Load Prediction model.

As shown in the above figure, initially the samples are applied to the forward direction and then to the backward direction simultaneously. Followed by which the temporal attention function is applied to obtain the final results. Gated Recurrent Unit comprises of two gate control functions, namely, an Update Gate (UG) and a Reset Gate (RG). On one hand the UG controls the extent to which the Antecedent Hidden State (AHS) of the corresponding cloud user (i.e. obtained via samples) is incorporated into the current input and on the other hand the RG ascertains the extent to which information of the corresponding cloud user (i.e. obtained via samples) takes control of an influence on the current state. Therefore GRU is said to be updated on the basis of the UG and RG of the current input corresponding cloud user (i.e. obtained via samples) and the AHS.

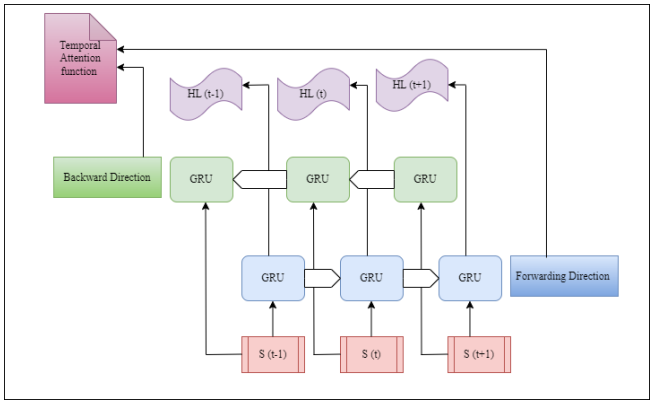

Figure 2: Structure of Attention Bidirectional Gated Recurrent Unit (ABiGRU) based Cloud Load Prediction model

Initially, the Update Gate (UG) result of current input (i.e. corresponding cloud user obtained via samples) is formulated as given below.

$$Inp_i = a_m S_t + b_m HL_{t-1} \quad (1)$$

$$UG_t = \sigma\left(Inp_i\right) \quad (2)$$

From the above equations (1) and (2) results the input information of the UG excitation function 'Inp_i' is arrived at based on the UG weights 'a_m', 'b_m' activated by the sigmoid activation function 'σ' for the corresponding hidden layer state (i.e. UG) 'HL' for the sample 'S' evolved at time 't'. Similarly, the Reset Gate (RG) result of current input (i.e. corresponding cloud user obtained via samples) is mathematically represented as given below.

$$Inp_j = a_n S_t + b_n HL_{t-1} \quad (3)$$

$$RG_t = \sigma\left(Inp_j\right) \quad (4)$$

From the above equations (3) and (4) results the input information of the RG excitation function 'Inp_j' is arrived at on the basis of the RG weights 'a_n', 'b_n' activated by the sigmoid activation function 'σ' corresponding to hidden layer state (i.e. RG) 'HL' for sample 'S' evolved at time 't'. The final output of GRU taking into account both the RG and UG is mathematically represented as given below.

$$HL_t = \left(1 - UG_t\right) HL_{t-1} + UG_t HL_t^{'} \quad (5)$$

From the above equation (5) the GRU final output 'HL_t' results are arrived at based on the updated value '$HL_t^{'}$' obtained by 'RG_t', 'HL_{t-1}' along with the current state. Nevertheless, the conventional GRU model has a low utilization rate of future workload forecasting. Hence, this work has enhanced the GRU model and presented a bidirectional GRU (BiGRU) model with the objective of predicting entire work load information via forward and backward prediction.

The forward calculation procedure for cloud work load prediction is similar as with the GRU model calculation while the reverse calculation procedure for cloud work load is mathematically represented as given below.

$$UG_t^p = \sigma\left(a_m^p S_t + b_m^p HL_{t+1}\right) \quad (6)$$

$$RG_t^p = \sigma\left(a_n^p S_t + b_n^p HL_{t+1}\right) \quad (7)$$

From the above equations (6) and (7) 'UG_t^p' and 'RG_t^p' denotes reverse gating results whereas 'a_m^p', 'b_m^p', 'a_n^p' and

'b_n^p' represents the subsequent weights. The bidirectional weighted integrated technique combines bidirectional HL states that input information from past cloud user request and future time instances of predicted point.

The integrated technique applied in this work is weighted sum and the evaluation procedure of the HL state is mathematically formulated as given below.

$$HL_t = l * HL_t^{for} + (1-l) HL_t^{back} \tag{8}$$

From the above equation (8) 'l' functions as weighted sum for the corresponding forward hidden layer 'HL_t^{for}' and backward hidden layer 'HL_t^{back}' placed by the cloud user request at time 't'.

$$Res_t = \sigma(HL_t * a_{Res}) \tag{9}$$

From the above equation (9) the future workload prediction result 'Res_t' is obtained based on the integrated function 'HL_t' and the weight from the hidden layer to the output layer 'a_{Res}'. The weight from the hidden layer to the output layer 'a_{Res}' is selected in such a manner so as to minimize the loss (i.e. difference between actual and predicted results) using the Stochastic Gradient Descent as given below.

$$a_{i+1} = a_i - \alpha * \nabla_a J(a, Obs_i, Pred_i),\ where \nabla_a J(a, Obs_i, Pred_i) = MSE \tag{10}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Obs - Pred)^2 \tag{11}$$

From the above equations (10) and (11) '$\nabla_a J(a, Obs_i, Pred_i)$' refers to the objective function that remains in minimizing the loss function using mean square error 'MSE', arrived at based on the observed results 'Obs' and predicted results '$Pred$'.

Finally for analyzing complicated temporal patterns, both in forward and backward directions Temporal Attention Mechanism is employed to concentrate on the most relevant parts of the input data (i.e. CPU usage and memory usage) at each time step for making prediction. Also employing Temporal Attention Mechanism permits to concentrate on the most relevant parts of the input data (i.e. examining energy usage over time to identify CPU usage and memory usage) while making cloud load predictions. The Temporal Attention Mechanism learns long time dependent characteristics of historical data (i.e. examining energy usage over time to identify CPU usage and memory usage) and allocates higher weights to more relevant input samples over a specific time period. Hence, we can lay hold off the principal complicated temporal patterns along with the most relevant parts of the input data by measuring destination weights and producing the attention model straight forward to interpret. This is mathematically represented as given below.

$$TAM = V(CPU, Mem).\sigma\left(\left(\left(S_i^{(l-1)}\right)^T W_1\right) W_2\left(W_3 S_i^{(l-1)}\right) + B\right) \tag{12}$$

$$TAM_{ij}^{'} = \frac{\exp(TAM_{ij})}{\sum_{j=1}^{Ch_{l-1}} \exp(TAM_{ij})} \tag{13}$$

Algorithm 1: Attention Bidirectional Gated Recurrent Unit based Cloud Load Prediction

Input: Dataset 'DS', Features '$F = \{F_1, F_2, \ldots, F_m\}$', Samples '$S = \{S_1, S_2, \ldots, S_n\}$',

Output: Optimal and Temporal Normalized Load Prediction results

1: **Initialize** '$m = 11$', '$n = 50000$', bias '$B = 0.01$'
2: **Begin**
3: **For** each Dataset 'DS' with Features 'F' and Samples 'S'
4: Formulate Update Gate (UG) result of current input (i.e. corresponding cloud user obtained via samples) according to (1) and (2)
5: Formulate Reset Gate (UG) result of current input (i.e. corresponding cloud user obtained via samples) according to (3) and (4)
6: Evaluate final output of GRU according to (5)
7: Evaluate bidirectional reverse calculation procedure for obtaining cloud work load according to (6) and (7)
8: Evaluate bidirectional weighted integrated function according to (8)
9: Obtain future workload prediction result according to (9)
10: Measure weight using Stochastic Gradient Descent according to (10) and (11)
11: Evaluate temporal attention matrix results according to (12) and (13)
12: **If** '$S[VM](CPU\_cores) + S[VM](Mem\_prov) \le TAM_{ij}^{'}$'
13: **Then** under load
14: **Go to** step 24
15: **End if**
16: **If** '$S[VM](CPU\_cores) + S[VM](Mem\_prov) > TAM_{ij}^{'}$'
17: **Then** over load
18: **Go to** step 24
19: **End if**
20: **If** '$S[VM](CPU\_cores) + S[VM](Mem\_prov) = TAM_{ij}^{'}$'
21: **Then** balanced
22: **Go to** step 24
23: **End if**
24: **Return** load results 'LR'
25: **End for**
26: **End**

From the above equations (12) and (13), the temporal attention matrix results '$TAM_{ij}^{'}$' are arrived at based on the values 'V' associated with each sample input (i.e. representing CPU usage and Memory usage obtained from the dataset) while bias 'B', '$W_1 \in \mathbb{R}^n$', '$W_2 \in \mathbb{R}^{Ch_{l-1}*n}$' and '$W_3 \in \mathbb{R}^{Ch_{l-1}}$' are said to be the learnable parameter for corresponding number of channels 'Ch_{l-1}' of sample input in the '$l-th$' layer. The pseudo code representation of Attention Bidirectional Gated Recurrent Unit based Cloud Load Prediction is given below.

As given in the above algorithm with the objective of ensuring optimal and temporal normalized load prediction results, the overall process is split into two parts. First, the input gwa-bitbrains dataset is subjected to cloud load prediction via Bidirectional Gated Recurrent Unit (BiGRU) and adaptive attention mechanism. Second, the BiGRU model extracts features constituting the association between sample input instances that in turn predict workload over time. To be more specific, the BiGRU model by extracting features highly dependent on input time series data via both forward and backward computational models. Employing BiGRU model for Cloud Load Prediction validate backtracking consolidation of virtual machine resulting in lower energy consumption. Also in the context of cloud load prediction, overhead refers to the resources (like CPU usage and memory usage) and time consumed during the prediction process probably influencing the overall effectiveness of the CC environment. With the intent of minimizing overhead or the time consumed in training the model, attention mechanism is then designed that by concentrating on the most relevant input data corroborates the objective.

Weight-adaptive Sparrow Search Optimization (WSSO) based fine-tuning

Once with the optimal load results obtained, an optimal load balancing model is designed in this section. For obtaining optimal load results, Weight-adaptive Sparrow Search Optimization (WSSO) algorithm is employed. The WSSO algorithm by fine-tuning the parameters of the ABiGRU model in turn makes certain accurate and optimal load balancing is ensured. The WSSO algorithm is applied to optimize ABiGRU model hyperparameters (i.e. learning rate), to enhance its prediction accuracy. The WSSO algorithm in our work is used for cloud load balancing by distributing cloud user requested tasks intelligently across virtual machines (VMs) based on their fitness, aiming to optimize resource utilization in an extensive manner. The attention weight in the ABiGRU model is fine-tuned via the Weight-adaptive Sparrow Search Optimization (WSSO) algorithm to accurately anticipate future workload in CC environment. Figure 3 shows the structure of Weight-adaptive Sparrow Search Optimization model.

As shown in the above figure, two types of sparrows (i.e. virtual machines), namely, producer and scrounger are said to exist. Excluding for the producer (i.e. virtual machines with high fitness value that actively search to accept cloud user request), all of the other sparrows are scroungers (i.e. virtual machines with low fitness values). The WSSO require to updates position of producer and the position of scrounger via producer's own fitness and it has the potentiality in guiding the broad search range of entire sparrow population. In the interest of getting better fitness, the scroungers will either follow the producers to forage or go to discrete locations to gain additional energy.

By simulating the cumulative behavior of entire sparrow population (i.e. virtual machine), WSSO algorithm seeks the optimal solution in the solution space (i.e. cloud computing environment). Let us assume that the sparrow (i.e. virtual machine) population 'VM' comprises of 'j' sparrow and each sparrow have 'i' dimension, then it is mathematically formulated as given below.

$$VM = \begin{bmatrix} vm_{11} & vm_{12} & .. & vm_{1m} \\ vm_{21} & vm_{22} & .. & vm_{2m} \\ .. & .. & .. & .. \\ vm_{n1} & vm_{n2} & .. & vm_{nm} \end{bmatrix} \tag{14}$$

The fitness value '$F(VM)$' for all sparrow or virtual machine is then mathematically represented as given below.

$$F(VM) = \begin{bmatrix} f(vm_{11}, vm_{12}, \ldots vm_{1m}) \\ f(vm_{21}, vm_{22}, \ldots vm_{2m}) \\ \cdots \\ f(vm_{n1}, vm_{n2}, \ldots vm_{nm}) \end{bmatrix} \tag{15}$$

From the above equation (15), '$f(.)$' denotes the fitness function. To address issue relating to unequal distribution of population (i.e. virtual machine) in the solution space (i.e. cloud computing environment) an adaptive weight is formulated 'ω' to ensure tradeoff between local and global search. The adaptive weight 'ω' is then mathematically represented as given below.

$$\begin{cases} x(t) = \exp\left(2 * \dfrac{1-\sin(\pi t)}{Max_{iter}}\right) - \exp\left(-2 * \dfrac{1-\sin(\pi t)}{Max_{iter}}\right) + \gamma \\ y(t) = \exp\left(2 * \dfrac{1-\cos(\pi t)}{Max_{iter}}\right) + \exp\left(-2 * \dfrac{1-\cos(\pi t)}{Max_{iter}}\right) + \delta \\ \omega(t) = x(t) * y(t) \end{cases} \tag{16}$$

From the above equation (16) the adaptive weight 'ω' result, is arrived at based on the maximum number

of iterations 'Max_{iter}' and two random numbers 'γ', 'δ' respectively. Then, the update position of the producer sparrow (i.e. virtual machine with high fitness value that actively search to accept cloud user request) after adding adaptive weight 'ω' is as given below.

$$VM_{ij}^{t+1} = \begin{cases} \left(VM_{ij}^{t} + \omega * \left(OF_{ij}^{t} - VM_{ij}^{t}\right)\right) * \gamma, \; if \; PL \le AL \\ \omega * VM_{ij}^{t}, \; if \;\; PL > AL \end{cases} \quad (17)$$

From the above equation result (17) 'OF_{ij}^{t}' forms the optimal fitness function in the current iteration if the results of prediction load 'PL' is less than or equal to the actual load 'AL' of the '$i-th$' producer sparrow in the '$j-th$' dimension for 't' number of iterations. Also, 'VM_{ij}^{t+1}' denotes the updated virtual machined selected by fine-tuning attention weight accordingly. The pseudo code representation of Weight-adaptive Sparrow Search Optimization (WSSO) based fine-tuning for cloud load balancing is given below.

As given in the above algorithm, the attention weight in the ABiGRU model is fine-tuned, with minimal makespan and improved throughput a weight adaptive function is introduced. First, virtual machines in the overall cloud computing environment are initialized. Second, the fitness value for each virtual machine in the overall cloud computing environment is formulated. With the reason that the initial population of the virtual machine is not distributed uniformly in the search space or CC environment, hence results in slow speed of convergence. To adapt to this unequal distribution an adaptive weight is formulated based on the sparrow search optimization function and accordingly obtains optimal strategy in fine-tuning the weight of ABiGRU model. This in turn not only reduces the makespan but also improves throughput considerably.

Performance evaluation

Simulation tool

The proposed method Attention Bidirectional Gated and Weight-adaptive Sparrow Search Optimization (ABiG-WSSO) is implemented throughout CloudSim simulation toolkit using gwa-bitbrains acquired from https://www.kaggle.com/datasets/gauravdhamane/gwa-bitbrains/data. The employment of this toolkit has received notable vogue among researchers and developers in the current cloud-related research domain. It efficiently minimizes the requirement need for and costs analogous with obtaining computing facilities for performance assessment and research modeling. This simulation tool acts as an external framework that can be straightforwardly downloaded and combined into JAVA programming environment. With the intent of performing simulation, the CC environment CloudSim was integrated into the Eclipse IDE for Java developers running the Windows 10 operating system.

Performance metrics

A set of metrics are utilized in measuring the algorithm efficiency such as makespan, energy consumption, throughput and overhead. These four performance metrics utilized in this approach are selected to put forward a complete evaluation of the effectiveness of the proposed method. Makespan in the context of cloud load balancing denotes the total time consumed in accomplishing all cloud user requested tasks. The main objective of load balancing algorithms remains in reducing this makespan time by effectively distributing workload across available resources. Energy consumption in cloud load balancing refers to the energy consumed while performing the process of distributing workloads across several servers, circumventing overloads and making certain significant utilization of resource.

Throughput refers to the amount of data being transferred per unit time in a successful manner. By distributing traffic across numerous cloud service providers throughput is said to be enhanced during the process of efficient load balancing. Finally, overhead refers to the resources consumed during the process of load balancing. Together, these four performance metrics, makespan, energy consumption, overhead and throughput provide a full view of the ability of the proposed method to manage resource allocation and workload distribution efficiently.

Energy consumption

Energy consumption denotes the sum of energy consumed by samples or cloud user physical machines. The energy

Algorithm 2: Weight-adaptive Sparrow Search Optimization based fine-tuning for cloud load balancing

Input: Dataset 'DS', Features '$F = \{F_1, F_2, \ldots, F_m\}$', Samples '$S = \{S_1, S_2, \ldots, S_n\}$',

Output: Optimal fine-tuned cloud load balancing

1: **Initialize** '$m = 11$', '$n = 50000$', '$\gamma, \delta \in [0,1]$'

2: **Initialize** random number '$RN = (0,1)$'

3: **Begin**

4: **For** each Dataset 'DS' with Features 'F', Samples 'S' and load results 'LR'

5: Formulate sparrow (i.e. virtual machine) population according to (14)

6: Formulate fitness value '$F(VM)$' for all sparrow or virtual machine according to (15)

7: Evaluate adaptive weight 'ω' according to (16)

8: Fine-tune attention weight according to (17)

9: **Return** optimal virtual machine selected 'VM_{ij}^{t+1}'

10: **End for**

11: **End**

consumption mathematical formulated is represented as given below.

$$EC_{i,l} = Pow_l EC_{T_i^j} \tag{18}$$

$$TEC_{i,l} = \sum_{i=1}^{n}\sum_{j=1}^{m} Pow_l EC_{T_i^j} \tag{19}$$

From the above equations (18) and (19) the total energy consumption '$TEC_{i,l}$' is measured based on the power 'Pow_l' and energy consumed by task 'T_i' executing on 'VM_l' '$EC_{T_i^j}$' respectively.

Overhead

Load balancers necessitate resources to process cloud user requesting tasks, manage connections and validate load balancing algorithms. In our work CPU and memory involved is taking into consideration.

Makespan

This performance metric denotes the maximum makespan among all virtual machines (VMs). The makespan performance metric provides insight into the overall effectiveness of the method in managing the completion times across distinct VMs. The makespan calculation is based on the maximum completion time of cloudlets for each VM as given in below equation.

$$MS\left(CT_{max}\right) = \max \sum_{i=1}^{n} T_{i,1}F_{i,1}, T_{i,2}F_{i,2}, \ldots, T_{i,l}F_{i,l} \tag{20}$$

$$F_{i,l} = \begin{cases} 1, if\ T_i \rightarrow VM_l \\ 0, Otherwise \end{cases} \tag{21}$$

From the above equations (20) and (21) makspan 'MS' is measured based on the maximum completion of last subtask 'CT_{max}' taking into consideration the allocation '$if\ T_i \rightarrow VM_l$' or non-allocation of virtual machine to a specific task respectively.

Throughput

Throughput evaluates the rate at which data or data packet is transmitted or received, denoting the effectiveness of the load balancer in controlling traffic. This is mathematically represented as given below

$$Th = \sum_{i=1}^{n} ET_i \tag{22}$$

The above four performance metrics, energy consumption, overhead, makespan and throughput were selected to comprehensively evaluate the efficiency of the proposed method. These metrics assess task completion times, management of resources and distribution of workload across VMs, imparting a flexible view of the efficiency of the proposed method in optimizing cloud load balancing.

Results and Discussion

Here, the complete evaluation of the proposed method Attention Bidirectional Gated and Weight-adaptive Sparrow Search Optimization (ABiG-WSSO) final outcome is illustrated. The performance along with the comparative evaluation employing Enhanced Dynamic Load Balancing (EDLB) and Deep learning with Particle Swarm Optimization and Genetic Algorithm (DPSO-GA) is performed to illustrate the efficiency of the work. By using Cloudsim, the proposed method and traditional methods, EDLB and DPSO-GA are executed using GWA-T-12 Bitbrains datasets are publicly available on the internet.

Performance analysis of energy consumption

In the first experiment, we assess the performance of our ABiG-WSSO method against a state-of-the-art method, the Enhanced Dynamic Load Balancing (EDLB) [1] and Deep learning with Particle Swarm Optimization and Genetic Algorithm (DPSO-GA) in terms of energy consumption. The ABiG-WSSO method employs a hybrid approach, combining the Attention Bidirectional Gated Recurrent Unit (ABiGRU) and Weight-adaptive Sparrow Search Optimization (WSSO) to enhance cloud load balancing efficiency in cloud computing environments. Table 3 given below lists the energy consumption results by substituting the values in equations (18) and (19) using ABiG-WSSO, EDLB [1] and DPSO-GA .

Energy consumption as illustrated in the above figure 4 provides insight into energy consumption efficiency per cloud user requested task, which is specifically paramount in scenarios with varying task complexities. By concentrating on this performance metric, we can measure how well

Table 3: Energy consumption measure using ABiGRU-WSSO, EDLB , DPSO-GA

Number of user requested tasks	*Energy consumption (Joules)*		
	ABiG-WSSO	*EDLB*	*DPSO-GA*
1000	32	41	49
2000	38	47	60
3000	45	54	57
4000	58	67	80
5000	65	74	87
6000	55	64	77
7000	48	57	70
8000	40	48	61
9000	52	61	74
10000	48	57	70

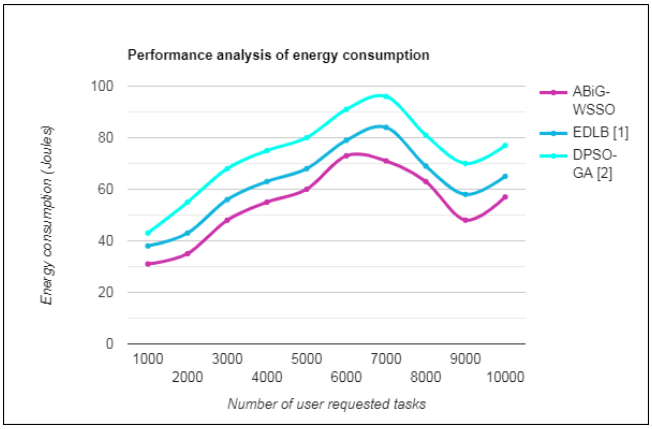

Figure 3: Graphical representations of the proposed ABiG-WSSO, EDLB and DPSO-GA concerning the energy consumption

Table 4: Overhead measure using ABiGRU-WSSO, EDLB and DPSO-GA

Number of user requested tasks	*Overhead*		
	ABiG-WSSO	*EDLB*	*DPSO-GA*
1000	0.5	0.63	0.68
2000	0.58	0.71	0.75
3000	0.62	0.75	0.8
4000	0.65	0.78	0.83
5000	0.68	0.79	0.84
6000	0.72	0.83	0.89
7000	0.74	0.86	0.91
8000	0.74	0.86	0.91
9000	0.73	0.84	0.88
10000	0.73	0.84	0.88

the method handles individual cloud user requested tasks, making certain high performance under different workloads. Figure 3 illustrates that the proposed ABiG-WSSO inevitably outperforms EDLB with an average energy consumption of 47Joules and DPSO-GA with an average energy consumption of 60Joules upon comparison to proposed ABiG-WSSO method with an average energy consumption of 38Joules respectively for 2000 cloud user requested tasks. This result underscores the proposed ABiG-WSSO method effectiveness in optimizing total energy consumption. The reason behind the energy consumption improvement using proposed ABiG-WSSO method is owing to the application of Attention Bidirectional Gated Recurrent Unit based Cloud Load Prediction algorithm. By applying this algorithm, both the forward and backward temporal dependencies are captured by concentrating on relevant information (CPU usage and memory usage) employing attention mechanisms, therefore minimizing the energy consumption extensively. The overall improvement in the average energy consumption compared to the EDLB method is 16% whereas the energy consumption compared to the DPSO-GA method is 30%.

Performance analysis of overhead

In the second experiment, the performance measure of overhead employing the proposed ABiG-WSSO method upon comparison to the state-of-the-art method, the Enhanced Dynamic Load Balancing (EDLB) and Deep learning with Particle Swarm Optimization and Genetic Algorithm (DPSO-GA) in performed. Table 4 given below lists the overhead results using ABiG-WSSO, EDLB and DPSO-GA.

Overhead is critical for assessing the efficiency of cloud load balancing algorithm in utilizing processing resources. Reducing overhead not only decreases operational costs but also improves throughput that is predominant in CC environments. As shown in figure 4, proposed ABiG-WSSO method exhibits more effective utilization of processing

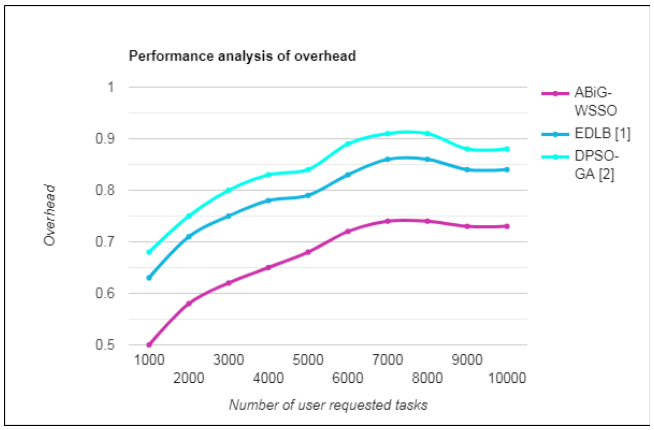

Figure 4: Graphical representations of the proposed ABiG-WSSO, EDLB and DPSO-GA concerning the overhead

resources. For instance, with 2000 cloud user requested tasks, ABiG-WSSO method records an overhead of 0.58, while EDLB and DPSO-GA 0.71 and 0.75, to a greater extent indicating the superiority of proposed ABiG-WSSO method in terms of overhead reduction. The reason for overhead minimization would to be contributed to the application of Attention Bidirectional Gated Recurrent Unit based Cloud Load Prediction model. By applying this model, Temporal Attention Mechanism learns long time dependent characteristics of historical data and accordingly assigns higher weights to more relevant input samples over a definite time period, therefore corroborating the objective of reduced overhead. The overall average improvement in overhead using proposed ABiG-WSSO method was 15% upon comparison to and 20% upon comparison to .

Performance analysis of makespan

In the third experiment, makespan performance measure is analyzed using three methods, ABiG-WSSO, Enhanced

Table 5: Makespan measure using ABiGRU-WSSO, EDLB and DPSO-GA

Number of user requested tasks	*Makespan (ms)*		
	ABiG-WSSO	*EDLB*	*DPSO-GA*
1000	31	38	43
2000	35	43	55
3000	48	56	68
4000	55	63	75
5000	60	68	80
6000	73	79	91
7000	71	84	96
8000	63	69	81
9000	48	58	70
10000	57	65	77

Table 6: Throughput measure using ABiGRU-WSSO, EDLB and DPSO-GA

Number of user requested tasks	*Throughput (bits/s)*		
	ABiG-WSSO	*EDLB*	*DPSO-GA*
1000	520	420	370
2000	690	480	400
3000	910	530	435
4000	1060	735	585
5000	2100	2023	2000
6000	2200	2055	2025
7000	2400	2235	2185
8000	2600	2315	2245
9000	2730	2415	2385
10000	3000	2535	2425

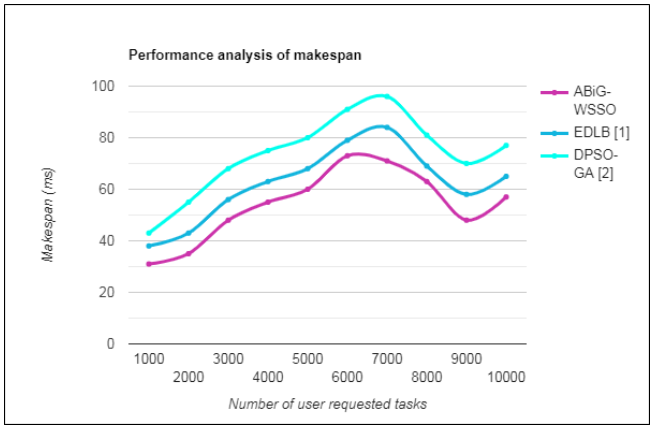

Figure 5: Graphical representations of the proposed ABiGRU-WSSO, EDLB and DPSO-GA concerning the makespan

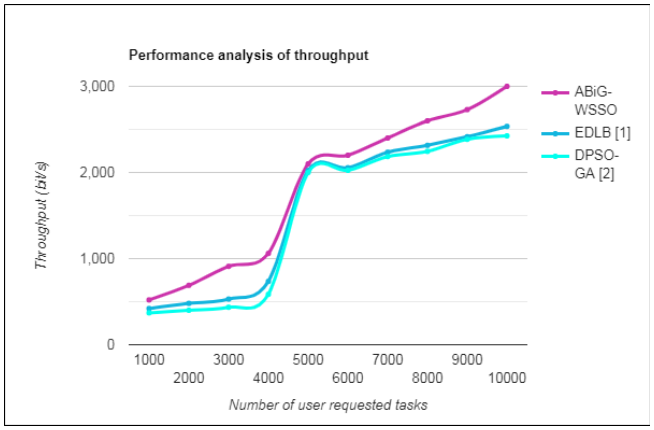

Figure 6: Graphical representations of the proposed ABiGRU-WSSO, EDLB and DPSO-GA concerning the throughput

Dynamic Load Balancing (EDLB) and Deep learning with Particle Swarm Optimization and Genetic Algorithm (DPSO-GA) . Table 5 given below list the makespan results by substituting the values in equation (22) employing ABiG-WSSO, EDLB and DPSO-GA .

Makespan as illustrated in the above figure 5 measures the effectiveness with which an algorithm controls or operate the entire set of cloud user requested tasks from start to finish. This performance metric straightforwardly influences the potentiality of the algorithm in minimizing the overall processing time, a critical component in CC environments where time effectiveness converts into enhanced satisfaction of cloud user. As illustrated in figure 6, the proposed ABiG-WSSO method constantly accomplishes a lower total makespan compared to EDLB and DPSO-GA across all number of cloud user requested tasks. For instance, with 1000 cloud user requested tasks, ABiG-WSSO method records a total makespan of 31ms, while EDLB logs 38ms and DPSO-GA logs 43ms highlighting a significant improvement in task completion time using ABiG-WSSO method. The reason was due to the application of Weight-adaptive Sparrow Search Optimization model. By applying this model, weights are fine-tuned in a dynamic or arbitrary manner during the optimization process, prospectively resulting in better convergence and exploration potentialities. This in turn minimizes total time required to accomplish all tasks. The overall improvement of the total makespan achieved 14% compared with EDLB and 27% compared with DPSO-GA .

Performance analysis of throughput

Finally in this section the detailed analysis of throughput is provided by making an elaborative detailing employing ABiG-WSSO, Enhanced Dynamic Load Balancing (EDLB) and Deep learning with Particle Swarm Optimization and Genetic Algorithm (DPSO-GA) . Finally, table 6 given below provides the throughput analysis results by substituting the values in equation (22) using ABiG-WSSO, EDLB and DPSO-GA .

Finally throughput shows in the above figure reflects how efficiently the algorithm influences available resources, with a higher percentage representing more efficient balancing of load in CC environment. As shown in Figure 6, proposed ABiG-WSSO maintains high throughput invariably. For instance, in scenarios with 10000 cloud user requested tasks, proposed ABiG-WSSO achieves a throughput rate of 3000 bits/s compared to EDLB and DPSO-GA 2535 bits/s and 2425 bits/s respectively. This inclination continues across various numbers of cloud user requested tasks confirming the robustness of proposed ABiG-WSSO in efficiently balancing load in CC environment. Even as the number of numbers of cloud user requested tasks increases, proposed ABiG-WSSO maintains steady rate of throughput. The reason behind the throughput improvement was owing to the application of Weight-adaptive Sparrow Search Optimization-based fine-tuning. By applying this algorithm, to adapt to the unequal distribution of initial population of the virtual machine in search space, an adaptive weight employing sparrow search optimization function was performed that in turn obtained fine-tuned the weight of ABiGRU model. The overall improvement of the proposed ABiG-WSSO method across number of cloud user requested task ranging from 1000 to 10000 was 24.54% upon comparison to and 38.09% upon comparison to .

Conclusion

In this paper, a novel method called Attention Bidirectional Gated and Weight-adaptive Sparrow Search Optimization (ABiG-WSSO) to accurately forecast future workload for scientific workflow scheduling in CC environment is proposed. This method targets makespan, energy consumption, throughput and overhead as multi objective parameters. The method composed in three parts, data collection was performed in first part, second part achieved minimal energy consumption and overhead using Attention Bidirectional Gated Recurrent Unit (ABiGRU) based Cloud Load Prediction model, in third part makespan and throughput improved optimal cloud load balancing was done using Weight-adaptive Sparrow Search Optimization-based fine-tuning model. To validate the efficiency of the proposed method, comparative experimental results are presented. The outcomes expressed that the proposed method performed better in terms of the multi objective performance metrics taken into consideration than the current methods EDLB and DPSO-GA using the publicly available gwa-bitbrains dataset.

Acknowledgement

The authors thank, DST-FIST, Government of India for funding towards infrastructure facilities at St.Joseph's College(Autonomous), Tiruchirappalli - 620 002.

References

Adaikalaraj, J. R., & Chandrasekar, C. (2023). To improve the performance on disk load balancing in a cloud environment using improved Lion optimization with min-max algorithm. *Measurement: Sensors, 27*. DOI: 10.1016/j.measen.2023.100834

Brahmam, M. G., & Anand, V. R. (2024). VMMISD: An efficient load balancing model for virtual machine migrations via fused meta-heuristics with iterative security measures and deep learning optimizations. *IEEE Access, 12*. DOI: 10.1109/ACCESS.2024.3373465

Devi, N., Dalal, S., Solanki, K., Dalal, S., Lilhore, U. K., Simaiya, S., & Nuris, N. (2024). A systematic literature review for load balancing and task scheduling techniques in cloud computing. *Artificial Intelligence Review, 57*. DOI: 10.1007/s10462-024-10925-w

Fan, Y. (2024, January). Load balance-aware dynamic cloud-edge-end collaborative offloading strategy. *PLOS ONE*. DOI: https://doi.org/10.1371/journal.pone.0296897

Farahi, R. (2025). A comprehensive overview of load balancing methods in software-defined networks. *Internet of Things, Discover*. DOI: https://doi.org/10.1007/s43926-025-00098-5

Gures, E., Shayea, I., Ergen, M., Azmi, M. H., & El-Saleh, A. A. (2022). Machine learning-based load balancing algorithms in future heterogeneous networks: A survey. *IEEE Access, 10*. DOI: 0.1109/ACCESS.2022.3161511

Han, H., Neira-Molina, H., Khan, A., Fang, M., Mahmoud, H. A., Awwad, E. M., Ahmed, B., & Ghadi, Y. Y. (2024). Advanced series decomposition with a gated recurrent unit and graph convolutional neural network for non-stationary data patterns. *Journal of Cloud Computing: Advances, Systems and Applications*. DOI: https://doi.org/10.1186/s13677-023-00560

Ikhlasse, H., Benjamin, D., Vincent, C., & Medromi, H. (2022). Multimodal cloud resources utilization forecasting using a bidirectional gated recurrent unit predictor based on a power efficient stacked denoising autoencoders. *Alexandria Engineering Journal, 61*. DOI: https://doi.org/10.1016/j.aej.2022.05.017

Kamila, N. K., Frnda, J., Pani, S. K., Das, R., Islam, S. M. N., Bharti, P. K., & Muduli, K. (2022). Machine learning model design for high performance cloud computing & load balancing resiliency: An innovative approach. *Journal of King Saud University – Computer and Information Sciences, 34*. DOI: 10.1016/j.jksuci.2022.10.001

Lahande, P. V., Kaveri, P. R., Saini, J. R., Kotecha, K., & Alfarhood, S. (2023). Reinforcement learning approach for optimizing cloud resource utilization with load balancing. *IEEE Access, 11*. DOI: 10.1109/ACCESS.2023.3329557

Lilhore, U. K., Simaiya, S., Sharma, Y. K., Rai, A. K., Padmaja, S. M., Nabilal, K. V., Kumar, V., Alroobaea, R., & Alsufyani, H. (2025). Cloud-edge hybrid deep learning framework for scalable IoT resource optimization. *Journal of Cloud Computing: Advances, Systems and Applications*. DOI: https://doi.org/10.1186/s13677-025-00729-w

Lohumi, Y., Gangodkar, D., Srivasatava, P., Khan, M. Z., Alahmadi, A., & Alahmadi, A. H. (2023). Load balancing in cloud environment: A state-of-the-art review. *IEEE Access, 11*. DOI: 10.1109/ACCESS.2023.3337146

Mathanraj, E., & Reddy, R. N. (2024). Enhanced principal component gradient round-robin load balancing in cloud computing. *The Scientific Temper, 15*. DOI: https://doi.org/10.58414/SCIENTIFICTEMPER.2024.15.1.32

Muneeswari, G., Madavarapu, J. B., Ramani, R., Rajeshkumar, C., & Singh, C. J. C. (2024). GEP optimization for load balancing of virtual machines (LBVM) in cloud computing. *Measurement:*

Sensors, 33. DOI: 10.1016/j.measen.2024.101076

Narsipuram, M., Rai, A., & Tiwari, A. (2024). Load balancing demystified: A comprehensive study of load balancing architectures in cloud computing. *Smart Internet of Things, 1*. DOI: https://doi.org/10.22105/siot.v1i2.36

Pradhan, A., Bisoy, S. K., Kautish, S., Jasser, M. B., & Mohamed, A. W. (2022). Intelligent decision-making of load balancing using deep reinforcement learning and parallel PSO in cloud environment. *IEEE Access, 10*. DOI: 10.1109/ACCESS.2022.3192628

Predić, B., Jovanovic, L., Simic, V., Bacanin, N., Zivkovic, M., Spalevic, P., Budimirovic, N., & Dobrojevic, M. (2023). Cloud-load forecasting via decomposition-aided attention recurrent neural network tuned by modified particle swarm optimization. *Complex & Intelligent Systems*. DOI: https://doi.org/10.1007/s40747-023-01265-3

Saini, H., Singh, G., Kaur, A., Saini, S., Wani, N. A., Chopra, V., Akhtar, Z., & Bhat, S. A. (2024). Hybrid optimization machine learning framework for enhancing trust and security in cloud network. *IEEE Access, 12*. DOI: 10.1038/s41598-024-78262-0

Sharma, A., Rani, S., & Driss, M. (2024). Hybrid evolutionary machine learning model for advanced intrusion detection architecture for cyber threat identification. *PLOS ONE*. DOI: https://doi.org/10.1371/journal.pone.0308206

Simaiya, S., Lilhore, U. K., Sharma, Y. K., Brahma Rao, K. B. V., Maheswara Rao, V. V. R., Baliyan, A., Bijalwan, A., & Alroobaea, R. (2024). A hybrid cloud load balancing and host utilization prediction method using deep learning and optimization techniques. *Scientific Reports*. DOI: 10.1038/s41598-024-51466-0

Sulimani, H., Sulimani, R., Ramezani, F., Naderpour, M., Huo, H., Jan, T., & Prasad, M. (2024). HybOff: A hybrid offloading approach to improve load balancing in fog environments. *Journal of Cloud Computing: Advances, Systems and Applications*. DOI: https://doi.org/10.1186/s13677-024-00663-3

Xu, M., Song, C., Wu, H., Gill, S. S., Ye, K., & Xu, C. (2022). esDNN: Deep neural network-based multivariate workload prediction in cloud computing environments. *ACM Transactions on Internet Technology, 22*. DOI: https://doi.org/10.1145/3524114

Yildirim, E., & Akon, A. (2023). Predicting short-term variations in end-to-end cloud data transfer throughput using neural networks. *IEEE Access, 11*. DOI: 10.1109/ACCESS.2023.3299311

Zhang, K., Guo, W., Feng, J., & Liu, M. (2021). Load forecasting method based on improved deep learning in cloud computing environment. *Scientific Programming*. DOI: https://doi.org/10.1016/j.egyai.2022.100198

Zhanuzak, R., Ala'anzy, M. A., Othman, M., & Algarni, A. (2024). Optimizing cloud computing performance with an enhanced dynamic load balancing algorithm for superior task allocation. *IEEE Access, 12*. DOI: 10.1109/ACCESS.2024.3508793

Zohora, M. F., Farhin, F., & Kaiser, M. S. (2024). An enhanced round robin using dynamic time quantum for real-time asymmetric burst length processes in cloud computing environment. *PLOS ONE*. DOI: https://doi.org/10.1371/journal.pone.0304517