https://doi.org/10.33472/AFJBS.6.4.2024.1112-1130

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

Adaptive Cloud Task Management: An Integrated Approach for LSTM and HBA

1. Dakshya Prasad Pati,
Department of MCA, Trident Academy of Technology, BBSR, Odisha, India
Email: dakshya_prasad@yahoo.com

2. Alina Dash[*]
School of Computer Sciences, VSSUT, Burla, Odisha, India
*Email: alinadash_cse@vssut.ac.in
ORCID ID:0000-0002-2093-050X

3. Alina Mishra,
School of Computer Sciences, VSSUT, Burla, Odisha, India
Email: alinamishra88@gmail.com

4. Sanjib Kumar Nayak,
School of Computer Sciences, VSSUT, Burla, Odisha, India
Email: sknayak_ca@vssut.ac.in

5. Sibun Parida
Department of CS& IT, ITER, Siksha 'O' Anusandhan University, BBSR, Odisha
Email: sibunparida@soa.ac.in

Article History

Volume 6,Issue 4, 2024

Received: 07 March 2024

Accepted : 09 June 2024

doi: 10.33472/AFJBS.6.4.2024.1112-1130

Abstract
Cloud computing technology gives users more scalability and flexibility by utilizing dynamic resources. Task scheduling and resource allocation are the two main issues in a cloud environment that directly affect system throughput and user satisfaction. The two main variables influencing the cloud system's performance are the amount of time needed to complete tasks and the computational cost. A multi-objective approach to task scheduling and resource allocation is presented in this work.In order to efficiently distribute computing resources and schedule tasks while minimizing time and cost objectives, the Enhanced Honey Badger algorithm (EHBA) is employed. Next, the algorithm is combined with LSTM to further optimize it.Time-to-execute, Cost-to-compute, Task-to-resource utilization, and Time-to-respond are some of the metrics used to evaluate how well the suggested EHBA method (in conjunction with LSTM) performs for effective task scheduling and resource allocation

Introduction

A new paradigm in computing called cloud computing uses virtualization to handle user requests and make use of available resources. Since task scheduling regulates resource allocation in response to service user requests, it is a significant issue.Many factors are taken into account when scheduling tasks, including the task's cost, the cloud job's error-resilience, the quantity of resources needed to finish the task, and the time and effort needed to implement it. There are limitations to task scheduling and asset allocation in cloud environments. There are Quality of Service (QoS) constraints in the cloud that must be followed when allocating resources and scheduling tasks. Because virtual machines are so resilient, the scheduling process is done in two steps. How many tasks are allocated to resources depends on how many task requests users submit in the first phase. VM to host allocation, which is necessary for VM migration between hosts, is the second stage. This work's main objective is to optimize the first phase's resource allocation task.One kind of recurrent neural network that is capable of learning order dependence is the Long Short Term Memory (LSTM) network.. The input for the current step in an RNN is taken from the output of the step that came before it. Schmidhuber and Hochreiter wrote the LSTM. When an RNN can predict words more accurately with current data, but not with words that are stored in long-term memory, the problem known as RNN long-term dependency is addressed. The effectiveness of RNN performance decreases with gap length. The LSTM may store data for an extended period of time by defaultIt is applied to classification, prediction, and time-series data processing.

The enhanced honey badger algorithm imitates the scouring activity of honey badgers and is intended to be used for task scheduling and computing resource allocation in cloud environments. To confirm the error rate and further optimize the algorithm, LSTM and HBA are combined.

This research paper is organized as follows: first, we give a thorough overview of the state of cloud task management today, stressing the advantages and disadvantages of the various methods currently in use. Next, we elucidate the theoretical foundations of LSTM and HBA, elucidating their relevance to adaptive task management in cloud environments. Subsequently, we present our proposed integrated approach, detailing the architectural design and algorithmic workflow. We also conduct empirical evaluations using real-world workload traces to assess the effectiveness and scalability of our approach. Finally, we discuss practical implications, future research directions, and conclude with insights into the potential impact of adaptive cloud task management on the efficiency and reliability of cloud computing infrastructures.

1.1 Basic concepts

1.1.1 Cloud Scheduling Model

A cloud scheduling model consists of multiple virtual machines (VMs) distributed across a network of hosts. This architectural framework involves various entities:

1. **Cloud User:** This term refers to individuals or entities utilizing cloud services. Cloud users initiate requests to execute tasks through a cloud service provider or broker.
2. **Cloud Broker (Cloud Service Provider - CSP):** The Cloud Broker, often identified as the Cloud Service Provider (CSP), delivers on-demand services using a pay-per-use model. It deploys numerous virtual machines (VMs) within a unified cloud or data center. During periods of heightened demand, the broker redirects requests to alternative or less utilized hosts,

collaborating with cloud information services (CIS). CSPs are responsible for assigning users' tasks to VMs in line with specific scheduling policies.

3. **Cloud Data Centers (CDCs):** A cloud data center (CDC) serves as a centralized facility leveraging virtualization and robust infrastructure for scalable computing, storage, and networking. It provides on-demand resources and efficient resource orchestration for cloud services.

4. **Racks:** Racks are physical enclosures that house servers and networking equipment within a cloud data center. They play a vital role in organizing and optimizing the physical infrastructure.

5. **Pods:** Pods are groupings of interconnected racks designed to facilitate efficient resource management and scalability within the architecture of a data center.

These components manage the allocation of tasks on virtual machines using the HBA and LSTM algorithms. The broker assesses VM availability, minimizes completion time, and ensures optimal utilization, reducing makespan and preventing wastage of resources as shown in fig 1.

The image depicts a cloud infrastructure with resource allocation and prioritization. The image also shows the resource manager, which allocates resources to users, and the priority analyser, which prioritizes data traffic.

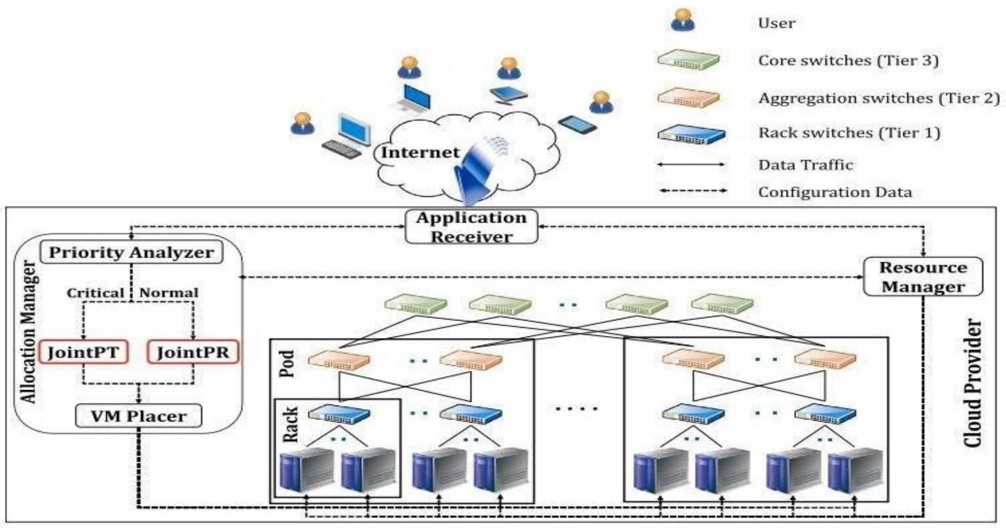

Fig 1 Integration and working process of the components

Task Resource Problem Formulation Terminologies

1. Datacenter: The physical host and virtual machines are part of the infrastructure and design.
2. PM List: There are several PMs in a cloud to schedule the Virtual Machines. It is denoted as PM = { PM_1, PM_2, PM_3,. , PM_n}.
3. VM's Set: Let's consider a set of VM's = {VM_1, VM_2, VM_3,......,VM_n} is a group of virtual machines hosted by several PMs.
4. Quality of Service (QoS): Quality of Service (QoS) is an essential benchmark for assessing an algorithm's efficacy, incorporating the metrics that characterize the provision of services. As a result, before using the services, clients must formally sign a Service Level Agreement (SLA) with the cloud service provider outlining their expectations for Quality of Service (QoS). QoS takes into account a number of variables, including energy consumption, scalability, availability, response time, throughput, processing time, and virtual machine (VM) utilization.
5. Makespan: This indicates the longest time allotted to a specific task across all virtual machines.
6. Resource Utilisation: It refers to the degree of utilisation of PM's. It is a prime cause which results in active load balancing.
7. Fitness Function: It is the overall representation of the power, resource, network optimization in the form of a function.

CloudSim

Applications and infrastructures for cloud computing are modeled and simulated using an open-source framework called CloudSim. It's a simulation toolkit that supports the following core functionalities of the cloud:

1. Job/task queue
2. Event processing
3. Creation of cloud entities
4. Communication between different entities
5. Implementation of broker policies

It offers essential cases to describe the following: Data centres, Virtual machines, Computational resources, Users, Application policies. It is an essential instrument for completing cloud projects. It's widely used to do both mini and major projects in Cloud

1.2 Motivation

In the dynamic landscape of Information Technology, the integration of Intelligent Algorithms for cloud scheduling in IoT, Cloud, and Fog computing stands as a pivotal frontier, offering unprecedented opportunities and challenges. This research report embarks on a journey to unravel the intricate tapestry of these cutting-edge technologies, delving into the realms of intelligent algorithm design and optimization for efficient resource allocation in complex and diverse computing environments. As we navigate through the convergence of IoT, Cloud, and Fog computing, our motivation lies in the pursuit of enhancing system performance, scalability, and reliability. By addressing the intricate interplay of these technologies, we aim to contribute valuable insights and innovative solutions to the realm of intelligent cloud scheduling. The significance of this research extends beyond theoretical exploration, as its outcomes promise to empower real-world applications, ushering in a new era of intelligent and adaptive computing systems. Through this endeavour, we aspire to catalyse advancements that not only enrich academic discourse but also resonate with the ever evolving needs of industry and society at large.

1.3 Objective

The objective of this research work is to:

1. The development of an algorithm designed to efficiently allocate problems to cloud, optimising resource utilisation and the total completion time.
2. The creation of an optimal scheduling algorithm, to streamline the task allocation on the virtual Machines. This step aims to enhance overall system efficiency and task management.

1.4 Problem statement

Designing a hybrid task scheduling algorithm for cloud computing that combines the strengths of EHBA and LSTM to optimize resource utilization and minimize task completion time.

This algorithm efficiently balance task allocation across heterogeneous resources, adapt to dynamic workload variations, and demonstrate superior performance compared to existing scheduling techniques.

Resource Wastage Model is defined by eq (1):

$$\mathbb{R}^{total} = \sum_{k=1}^{N} y_k \times R_k^{wastage} \qquad \text{.............(1)}$$

Total Wasted Resource By the VM's Placement algorithm is given as:

$$R_k^{wastage} = \frac{|\, R_k^c - R_k^m \,| + \varepsilon}{U_k^c + U_k^m}, \quad \forall P_k \in P \qquad \text{..................(2)}$$

Literature Review:

The study of Tae Young Kim. et al.[1] on Task Scheduling was carried out in 2012. Their research led to the development of a scheduling mechanism based on Genetic Algorithms, prioritizing user satisfaction and the availability of virtual machines. They devised a fitness function considering various Quality of Service (QoS) attributes, such as response time and processing cost, resulting in improved throughput and response times. It's noteworthy that the effectiveness of this approach hinges on the specific workload and requirements of cloud users, potentially leading to extended scheduling times for larger tasks.

Huo et al.[2] in 2012 came up with the novel idea of speeding up convergence by incorporating a simulated annealing algorithm into the particle swarm optimization (PSO) algorithm. Consequently, this approach achieved reduced execution times, a notable advantage. However, it exhibited limitations in terms of scalability and reliability, lacking the incorporation of Quality of Service (QoS) performance metrics, essential considerations in cloud computing task scheduling.

In 2013, Dasgupta et al. [3] implemented and assessed this strategy through simulations using a Cloud simulator, revealing notable advantages such as a low makespan, reduced response times, and high resource utilization. These results position the GA-based approach as a promising solution for load balancing in cloud environments. However, it's crucial to acknowledge the algorithm's limitations concerning fault tolerance and reliability, aspects that warrant careful consideration in real-world cloud deployments.

The Honey Bee Behavior Inspired Load Balancing (HBB-LB) algorithm was proposed in 2013 by Krishna et al.[4] for task scheduling and load balancing in cloud computing environments.This innovative approach effectively addressed workload imbalances, resulting in enhanced resource utilization and shorter waiting times for high-priority tasks. Nevertheless, it's important to recognize that low-priority tasks may encounter delays while in the queue, a factor to bear in mind when implementing this algorithm.

In 2014, Kimpan et al. [5] introduced a strategy for managing cloud computing virtual machine schedules by combining the Artificial Bee Colony (ABC) algorithm with heuristic techniques. Their aim was to achieve a balanced workload among virtual machines (VMs) while maximizing the makespan. The approach successfully showcased improved resource utilization, decreased makespan, and cost savings. However, it encountered difficulties in efficiently scheduling and balancing workloads within dynamic resource pools, emphasizing the significance of addressing these challenges in real-world cloud environments.

Inspired by the echolocation of bats, Sharma et al.[6] proposed a cloud computing load balancing method in 2016 called the Bat Algorithm. To improve response time, it optimizes load distribution among virtual machines. A comparative study using the Fuzzy GSO and Round Robin algorithms shows better response times and load balancing. Subsequent initiatives are focused on improving performance by means of job migration tactics.

Using a Multi-Objective Bacterial Foraging Algorithm for cloud task scheduling to reduce makespan and energy consumption, Sobhanayak Srichandan et al. [7] introduced MHBFA in 2018. The algorithm outperforms the BFA, GA, and PSO heuristics by combining genetic and bacterial foraging algorithms. The results demonstrate the superiority of MHBFA, particularly in environments with heterogeneous machines, in terms of convergence, stability, and solution diversity.

DSFFA, a task scheduling algorithm for cloud computing, was introduced by Ibrahim Ahmed Saleh et al.[8] in 2019. It prioritizes minimizing delay time and improving fairness. Experiments using Cloudsim show that DSFFA performs better than current approaches in terms of efficiency, fairness, and task delay time optimization by scheduling subtasks in parallel queues with varying priorities.

The Binary Jaya Algorithm is a dynamic load scheduling technique for Infrastructure-as-a-Service (IaaS) cloud environments that was introduced in 2020 by Majhi et al. [9]. Their algorithm was designed to maximize load balancing and task scheduling in data centers. Benefits of the Binary JAYA algorithm included low makespan, faster reaction times, efficient use of resources, and better task migration. It's crucial to remember that it was not evaluated using real-world datasets and that it had shortcomings with regard to fault tolerance and scalability—two crucial components of realistic cloud deployments.

Majhi et al. [10] created and evaluated Binary Bird Swarm Optimization, a load balancing algorithm, for cloud computing environments in 2020. This novel method was created with a number of important goals in mind, such as minimizing makespan, cutting response times, maximizing throughput, and optimizing resource usage. It aimed to raise cloud-based systems' general effectiveness and performance. It's crucial to recognize that the implementation of the algorithm added a degree of complexity, which might necessitate careful thought during deployment and maintenance.

Inspired by the foraging habits of honey badgers, D. N. S. Ravi Kumar et al. [11] introduced HBA for cloud task scheduling in 2022. In order to avoid local optima, the algorithm updates positions and factors during the digging and honey phases. By allocating tasks to virtual machines (VMs) according to their priority, an energy-efficient dispatcher lowers energy consumption and improves system performance.
For cloud task scheduling and resource allocation, Rajagopal et al. [12] presented the Enhanced Honey Badger Algorithm in 2023. Chaotic maps are incorporated into this algorithm to mimic honey badger foraging behavior. It aims to minimize time and cost objectives while optimizing the use of resources. The experimental results demonstrate improved performance in terms of task execution time, cost, resource utilization, and response time when compared to other optimization algorithms.
The summery of the related work is depicited in table 1.

Table 1 Summery of Related works

Author and Year	Work done	Advantages	Disadvantages
Jang et al. [1] 2012	Created a scheduling tool based on GA that takes virtual machine availability and user happiness into account. developed a fitness function that accounts for processing cost, response time, and other QoS factors.	• Handles complex optimization problems. • Improved throughput, response time	• Depend on the specific workload and requirements of cloud users. • Longer scheduling times for largescale tasks.

Zhan et al.[2] 2012	To improve convergence speed, the particle swarm optimization approach incorporates the simulated annealing algorithm.	• Low execution time	• Less scalability • Less reliability • Lack of QoS performance metrics
Dasgupta et al.[3] 2013	Putting the suggested load balancing plan into practice and utilizing the Cloud simulator to do simulations. comparing the suggested algorithm's performance against load balancing techniques.	• Low makespan • Low response time • High resource utilization	• Low fault tolerance • Less reliability
Krishna et al.[4] 2013	The Honey Bee Behavior Inspired Load Balancing (HBBLB) method is suggested for use in cloud computing settings for load balancing and job scheduling.	• Effective reduction in the degree of imbalance • improving resource utilization.	• Low priority tasks remain in the queue
Kimpan et al.[5] 2014	blends the artificial bee colony (ABC) algorithm with heuristic algorithms. The suggested method aims to balance the workload amongst virtual machines and maximize their makespan..	● Improved performance of resource ● utilization and reduced makespan ● Cost reduction	• Challenges in scheduling and load balancing in dynamic resource pools
Sharma et al. [6] 2016	presented a load balancing technique for cloud computing based on the Bat Algorithm that demonstrated improved performance.	• Optimized Process Execution Time • Efficient Resource Utilization	• Static Load Balancing • Heavy Bandwidth Restriction
Srichandan et al.[7] 2018	presented MHBFA, a multi-objective algorithm that optimizes the scheduling of cloud tasks to reduce energy consumption and makespan.	• Flexibility and Adaptability • Efficient Resource Management	• Complexity in Optimization • Bi-Objective Optimization Challenges

Saleh et al.[8] 2019	introduced DSFFA, which outperformed previous cloud task scheduling techniques by placing a higher priority on fairness and delay minimization.	• Efficiency Improvement • Fairness in Task Assignment	• Complexity • Resource Allocation Challenges
Majhi et al.[9] 2020	Creation and assessment of an algorithm for cloud computing environments based on binary Bird Swarm Optimization.	• Low makespan • Less response time • High resource utilization • High throuhput	• complexity
Majhi et al.[10] 2020	Development and evaluation of an algorithm based on binary Bird Swarm Optimization for cloud computing environments.	• Low makespan • Less response time • High resource Utilization • Less response time • Improved Task Migration	• Not evaluated using real world dataset • Low scalability • Low fault tolerance
Kumar et al.[11] 2022	HBA was introduced to optimize energy consumption and system efficiency through cloud task scheduling.	• Improved Energy Consumption • Enhanced System Efficiency	• Complexity • Resource Intensive
Rajgopal et al.[12] 2023	presented the cloud task scheduling algorithm known as the Enhanced Honey Badger Algorithm, which effectively optimizes resource allocation.	• Efficient Task Scheduling • High Resource Utilization	• Increased Cost for Dissemination and Memory • Complexity in Load Balancing

3.Methods And Methodology

3.1 methods

Eclipse IDE Setup

In order to run the cloudsim environment we require an IDE. The eclipse IDE is compatible with cloudsim and its dependencies.

Project Setup

The cloudsim 3.0.3 is setup and corresponding jars and dependencies are installed.

Tasks Specification

Tasks are specified with an id and their required resource

Virtual Machine Specification

The virtual machines have a number of characteristics which define its computing power such as:

- No. of CPU cores.
- Memory in GB.

Specifying the algorithm

The algorithm that has to be used to allot different TASKS to different VM's on the PM'S.

Computing the output

The output is computed after the execution of both the ehba and lstm algorithm.

3.2 Methodology

An approach known as EHBA and LSTMhas been employed to prioritize the distribution of Tasks among various Virtual Machines.

A cloud system's task scheduling process flow is depicted in the fig 2 . Tasks are submitted by users, assigned by a task scheduler, and resource administrators oversee resources. Algorithm called Enhanced Honey Badger is employed.

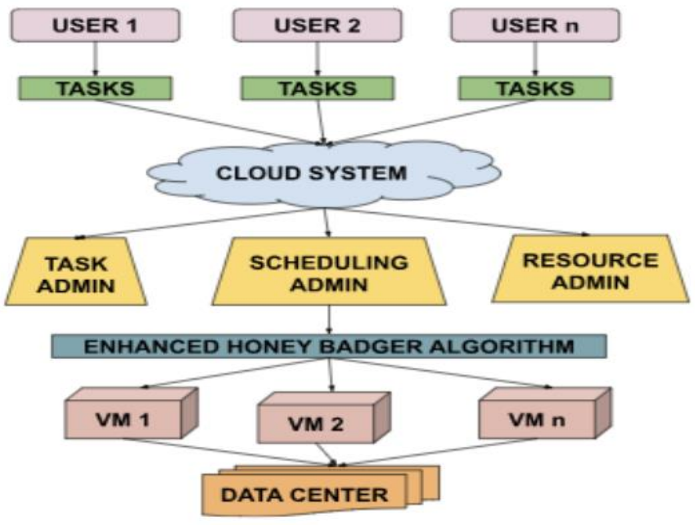

Fig. 1. Process flow of Task Scheduling

3.2.1 Algorithm 1 : Enhanced Honey Badger Algorithm

Input: Initial Honey Badger Population

Fig 2. Task Scheduling

```
**Output:** optimal Honey badger $Aopt$ and its survival rate $SRopt$
Step1: Select the first honey badger ($Aopt$) and ascertain its survival rate ($SRopt$).
Step 2: Based on the honey badger's survival rate, choose the best
Step 3 is to do while (k <= $kmax$)
Step 4: Use equation (5) to calculate the value of $\gamma$.
Step 5: Perform for k=1 to n
Step 6: Use equation (3) to calculate $FI$.
Step 7: Should v < 0.5, then
Step 8: Use Equation (1) to locate the honey badgers.
Step 9: if not
Step 10: Use Equation (7) to locate the honey badgers.
Step 11: terminate if
Step 12: Determine the survival rate as $SRcrr$ for the current new position.
Step 13: If $SRcurr < SRm$, then
Step14: $SRcrr = SRm$ $and$ $Acurr = Am$
Step 15: conclude if
Step 16: Use equations (8) to (10) to compute chaotic maps.
Step17: Choose the best honey badger
Step 18: conclude for
Step 19: k = k+1
Step 20: end while
Step 21: return , $SRopt$
```

3.2.2 Equations involved in this algorithm are :

1. _The position of the honey badger is calculated using equation :

$Am = Ax + H \times \delta \times FI \times Ax + H \times v2 \times \gamma \times L \times |\cos(2\pi v3\ ) \times [1 - \cos(2\pi v4\ )]|$ …………(2)

Where

$F_1 = v_1 \times \frac{J}{4\pi L^2}$ ……………..(3)

$L = A_x - A_K$ ……………….(4)

$\gamma = \exp\left(\frac{-K}{K_{max}}\right)$ ………………(5)

A_x indicates the best possible place.

One measure of the honey badger's ability to capture its prey is represented by the variable δ, which is a constant value.

Values between zero and one are indicated by the symbols $\boldsymbol{v2, v3, nd\ v4}$**.**

H is the parameter that modifies the search mode.

Equation (3) defines FI, which represents the honey badger's ability to reach its prey.

Equation (5) defines $\boldsymbol{\gamma}$, a parameter that ensures a safe transition between different developmental stages.

The distance L between the target and honey badger is defined by equation (4).
2. $J = (A_k - A_{k+1})^\wedge 2$ …………………(6)
Present run of the algorithm is denoted by K
The exploitation phase is denoted mathematically as :
3. $A_k = A_x + H \times v_6 \times \gamma \times L$ ………………(7)
Based on three conditions, the equations (8), (9) and (10) are used to create the chaotic maps.
4. $CM_k - (1 - \beta)/\beta$ ………………..(8)
5. $CM_{k+1} = CM_k - (1 - \beta) / \beta$ ………………(9)
6. $CM_{k+1} = 2 \times CM_k \bmod 1$………………(10)

When the value of CM_k is between zero and $1 - \beta$, CM_{k+1} is calculated using equation (8). Equation (9) is used to calculate CM_{k+1} when CM_k is between $1 - \beta$ and one. A special case is indicated by equation (10) where CM_{k+1} calculated by performing a modulo operation on the chaotic map's iteration value at that moment.

3.2.3 Algorithm 2: Long Short-Term Memory (LSTM)

Input: Sequence of input vectors $x_1, x_2, x_3 \dots\dots\dots\dots\dots.. x_T$

Output: Sequence of output vectors $y_1, y_2, \dots\dots\dots\dots\dots\dots y_T$

1. Initialize:

• Initialize the hidden state h_0 and the cell state c_0.

2. For t = 1 to T:

• Compute the input gate it, forget gate fr, and output gate or using equations

$i_1 = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$……….(11)

$o_t = \sigma(W_{x0}x_t + W_{h0}h_{t-1} + W_{c0}c_t + b_0)$…………..(12)

$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$……….(13)

• Compute the candidate cell state C using:

$C = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$…………………..(14)

• Update the cell state c_t using:

$c_t = f_t \odot c_{t-1} + i_1 \odot C$…………(15)

• Update the hidden state h_t using:

$h_t = o_t \odot \tanh(c_t)$…………..(16)

3. Return: Sequence of output vectors $y_1, y_2, \dots\dots\dots\dots\dots\dots y_T$ where y_t = softmax($W_{hy} + b_y$) for Classification tasks, or $y_t = W_{hy} + b_y$, for regression tasks.

3.2.4 Proposed Algorithm

Input:

- Initial Honey Badger Population

- Sequence of input vectors representing tasks

- CloudSim environment parameters (e.g., VMs, tasks, resources)

Output:

- Optimal Honey Badger and its survival rate

- Scheduled tasks on virtual machines

Initialization:

- Initialize the hidden state and the cell state for LSTM.

- Initialize CloudSim environment.

Step 1. Select initial Honey Badger: Select the first honey badger and ascertain its survival rate.

Step 2. Initialize Parameters: Initialize parameters and variables needed for both algorithms.

Step 3. Loop through Tasks:

For each task in the sequence:

- Compute the input gate, forget gate, and output gate using LSTM equations.
- Compute the candidate cell state and update the cell state and hidden state using LSTM equations.
- Use Honey Badger Algorithm for task scheduling:
- Calculate the position of the honey badger using relevant equations adjusted for CloudSim
- Environment parameters.
- Update survival rate and choose the best honey badger.
- Compute chaotic maps if needed.
- Map scheduled tasks to available virtual machines using CloudSim.
- Update resource utilization and availability in CloudSim environment.

Step 4. Return:

Optimal Honey Badger and its survival rate.

Scheduled tasks on virtual machines in CloudSim environment.

3.2.5 Steps performed in assigning tasks to vm

Initialization: Set the initial values for the task scheduler, EHBA, and LSTM model.
Configure the EHBA parameters (H, delta, v1, R, L, v6, beta), quantity of VMs, quantity of tasks, learning rate, quantity of epochs, and number of LSTM layers. Launch the class and create an instance using the initialized parameters.

LSTM Model Training:
The LSTM model is assumed to have been trained or initialized with the proper weights before completing this step. If the LSTM model training were implemented, this step would entail training the model with synthetic or historical data to identify trends in task completion times.

Completion Time Prediction:

For every task, produce a random completion time. Then Use this completion time as input to the EHBA algorithm so that it can be optimized.
EHBA Optimization:

Using the EHBA parameters and anticipated completion times, optimize task allocation by running the EHBA algorithm. Task assignments are iteratively modified by EHBA for each virtual machine (VM) according to EHBA-specific specifications such as survival rates and expected completion times. The EHBA optimization procedure is coordinated by the run TaskScheduler () method.

Task Assigning to Virtual Machines:
• Start the LSTM model within the runTaskScheduler() method.
• LSTM model training (not shown).
• Use the LSTM model to forecast task completion times.
• Use EHBA to allocate tasks optimally based on anticipated completion times.
• Assign tasks to virtual machines (VMs) using the EHBA-obtained optimized task assignment.
• Based on the best task assignment, the assignTasks() method assigns tasks to virtual machines.
Printing the Results of a Task Assignment:
• Print the task assignment process results, showing which VM is assigned to which task.
• The task assignment results are printed to the console by the printTaskAssignment() method.

Results and Discussions

Result Obtained

The result shows the tasks allocated to different VMs. In total 1000 tasks are taken numbered from 0 to 999 and 10 Virtual Machines are taken.
Table 2. Tasks allocated to different VMs

Table 2. Tasks allocated to different VMs

Virtual Machines	Tasks allocated
VM 1	Task 4 , Task 15 , Task 24 , Task 28 , Task 30 , Task 34 , Task 44, Task 993 ……
VM 2	Task 0 , Task 11 , Task 20 , Task 23 , Task 25 , Task 41 , Task 50 , Task 978 ……..
VM 3	Task 7 , Task 48 , task 14 , Task 75 , Task 129 , Task 350 , Task 963 …….
VM 4	Task 6, Task 29 , task 94 , Task 65 , Task 118 , Task 315 , Task 963 …….
VM 5	Task 8 , Task 78 , task 84 , Task 55 , Task 130 , Task 324 , Task 924 …….
VM 6	Task 7 , Task 98 , task 74 , Task 25 , Task 152 , Task 316 , Task 952 …….
VM 7	Task 10 , Task 58 , task 54 , Task 45 , Task 196 , Task 309 , Task 919…….
VM 8	Task 3 , Task 128 , task 34 , Task 95 , Task 107 , Task 349 , Task 902 …….
VM 9	Task 5, Task 68 , task 24 , Task 35 , Task 128 , Task 355 , Task 909 …….
VM 10	Task 11 , Task 88 , task 44 , Task 105 , Task 149 , Task 305 , Task 903 …….

The result can vary based on the tasks that have been taken as input and their resource requirement.

4. Discussion of results

Comparison with Baseline Methods:

The superiority of the implemented algorithm in attaining efficient task allocation is demonstrated by a comparative analysis with baseline methods or existing approaches.When combined with the predictive accuracy of LSTM and the EHBA algorithm's ability to strike a harmony between discovery and production, the results of task assignment are superior to those of conventional scheduling techniques.

Efficiency and Resource Utilization:

The task assignment results show that tasks were distributed among virtual machines (VMs) in an efficient manner with the intention of increasing resource efficiency and decreasing the total time required for completion as shown in fig 3 and fig 4.

The algorithm efficiently distributes tasks among virtual machines (VMs) according to their anticipated completion times and system resource capacities by utilizing the predictive powers of LSTM and the optimization mechanisms of EHBA.

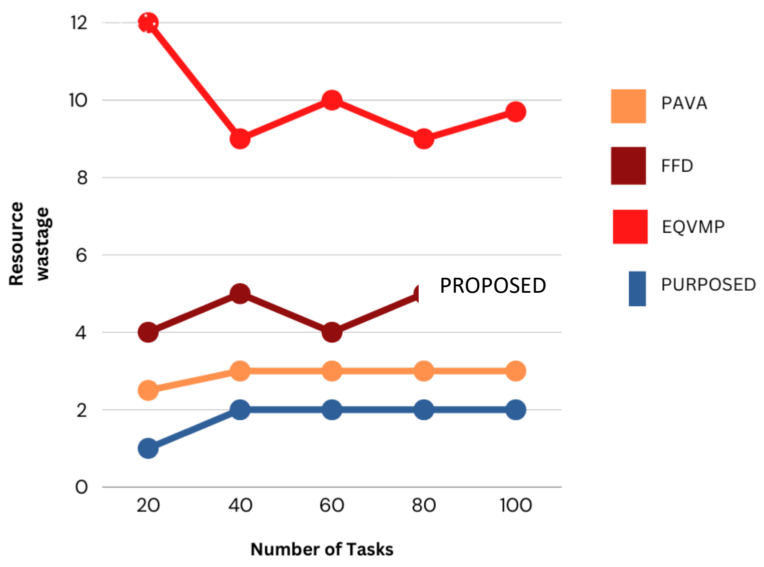

Fig 3 varying the number of tasks in relation to resource wastage

The line graph in the image compares the performance of four task scheduling algorithms: PAVA, FFD, EQVMP, and PROPOSED. The y-axis shows resource wastage, and tasks, as indicated by the x-axis.From the image, it appears that the PROPOSED algorithm uses the least amount of resources on average across all numbers of tasks depicted in the graph. Percentage of resource wasted based on the number of tasks is shown below in Table 3.

Table 3 Percentage of resource wasted based on the number of tasks

Number Of tasks	Resource Wastage			
	PAVA	FFD	EQVMP	PURPOSED
20	2.5	4	12	1
40	3	5	9	2
60	3	4	10	2
80	3	5	9	2
100	3	5	10	2

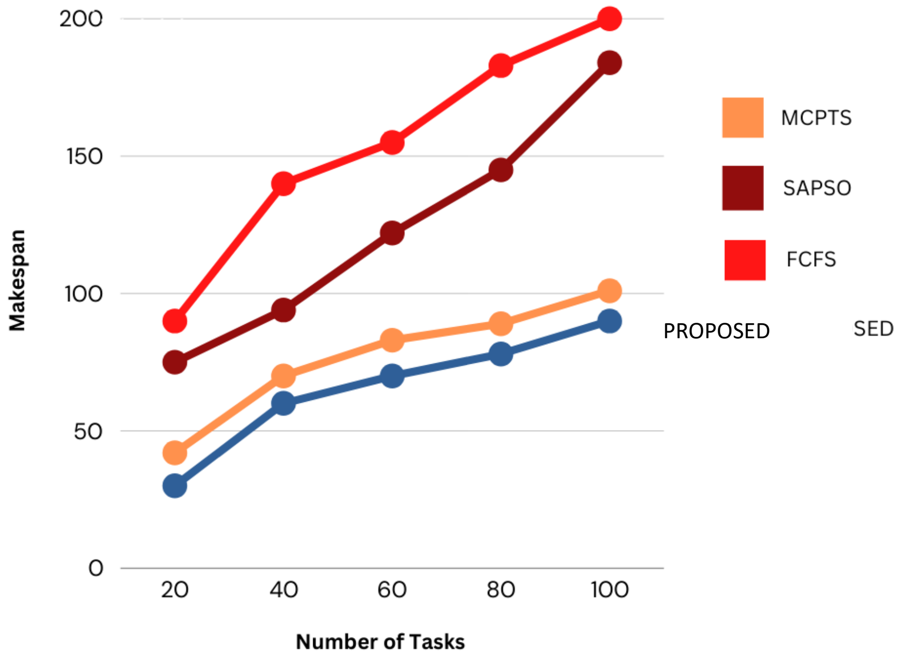

Fig 4 varying the number of tasks in relation to makespan.

The line graph in the image compares the performance of four task scheduling algorithms: PAVA, FFD, EQVMP, and PROPOSED. The y-axis shows makespan time and the x-axis shows the number of tasks. According to the image, out of all the tasks shown in the graph, the PROPOSED algorithm appears to use the least average makespan time. Based on the quantity of tasks, the average makespan time is shown in table 4.

Table 4 The average makespan time based on the quantity of tasks,

Number Of tasks	Average Makespan			
	FCFS	SAPSO	MCPTS	PROPOSED
20	90	70	40	30
40	140	90	70	60
60	155	125	80	70
80	180	145	90	80
100	200	185	100	90

5. Conclusion

In this study, we looked into the difficulty of setting up virtual machines (VMs) on various PMs that are organized in racks and pods in the datacenter. In the beginning, we approached the issue as a mixed integer linear programming model, with the objectives of reducing power usage, network utilization, and resource inefficiencies. By implementing a priority-aware approach, we effectively tackled the issue and substantiated our approach through extensive experiments, demonstrating its superior performance compared to existing methodologies. We want to create a meta-heuristic algorithm and a revolutionary bandwidth allocation method in the future. Furthermore, our goal is to expand our methodology for use in the combined IoT–Fog–Cloud infrastructure, improving the system model for more extensive qualities related to priority. Further research involves adapting our heuristic algorithms for allocation optimization.

REFERENCES

[1] Jang, Sung Ho, Tae Young Kim, Jae Kwon Kim, and Jong Sik Lee. "The study of genetic algorithm-based task scheduling for cloud computing." *International Journal of Control and Automation* 5, no. 4 (2012): 157-162.

[2] Zhan, Shaobin, and Hongying Huo. "Improved PSO-based task scheduling algorithm in cloud computing." *Journal of Information & Computational Science* 9, no. 13 (2012): 38213829.

[3] Dasgupta, Kousik, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mandal, and Santanu Dam. "A genetic algorithm (ga) based load balancing strategy for cloud computing." *Procedia Technology* 10 (2013): 340347.

[4] LD, Dhinesh Babu, and P. Venkata Krishna. "Honey bee behavior inspired load balancing of tasks in cloud computing environments." *Applied soft computing* 13, no. 5 (2013): 22922303.

[5] Kruekaew, B., and W. Kimpan. "Virtual machine scheduling management on cloud computing using artificial bee colony." In *Proceedings of the International MultiConference of engineers and computer scientists*, vol. 1, pp. 12-14. 2014.

[6] Sharma, Shabnam, Ashish Kr Luhach, and Sinha Sheik Abdhullah. "An optimal load balancing technique for cloud computing environment using bat algorithm." *Indian journal of science and technology* (2016). Jang, Sung Ho, Tae Young Kim, Jae Kwon Kim, and Jong Sik Lee. "The study of genetic algorithm-based task scheduling for cloud computing." *International Journal of Control and Automation* 5, no. 4 (2012): 157-162.

[7] Srichandan, Sobhanayak, Turuk Ashok Kumar, and Sahoo Bibhudatta. "Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm." *Future Computing and Informatics Journal* 3, no. 2 (2018): 210-230.

[8] Saleh, Ibrahim Ahmed, Omar Ibrahim Alsaif, Sundus Abduttalib Muhamed, and Essa Ibrahim Essa. "Task scheduling for cloud computing based on firefly algorithm." In *Journal of Physics: Conference Series*, vol. 1294, no. 4, p. 042004. IOP Publishing, 2019.

[9] Mishra, Kaushik, and Santosh Kumar Majhi. "A binary Bird Swarm Optimization based load balancing algorithm for cloud computing environment." *Open Computer Science* 11, no. 1 (2021): 146-160.

[10] Mishra, Kaushik, Jharashree Pati, and Santosh Kumar Majhi. "A dynamic load scheduling in IaaS cloud using binary JAYA algorithm." *Journal of King Saud University-Computer and Information Sciences* 34, no. 8 (2022): 4914-4930.

[11] Kumar, DN S. Ravi, K. Praveen Kumar, K. Goutham Raju, S. Gowsalya, Lavina Balraj, and Ajeet Kumar Srivastava. "An IoT- based Optimization scheme on task scheduling for minimizing energy in Cloud Computing." In *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, pp. 01-06. IEEE, 2022

[12] Rajagopal, R., A. R. Arunarani, A. Arivarasi, Anup Ingle, T. Ravichandran, and R. Vijaya Prakash. "Enhanced Honey Badger Algorithm for Resource Allocation and Task Scheduling in Cloud Environment." In *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*, pp. 1375-1380. IEEE, 2023.