



International Journal of Cloud Computing

ISSN online: 2043-9997 - ISSN print: 2043-9989

<https://www.inderscience.com/ijcc>

An optimised AI-driven swarm-based enhanced task scheduling model for cloud computing environment

Surinder Kaur, Jaspreet Singh, Vishal Bharti

DOI: [10.1504/IJCC.2025.10069555](https://doi.org/10.1504/IJCC.2025.10069555)

Article History:

Received:	13 July 2024
Last revised:	27 November 2024
Accepted:	20 December 2024
Published online:	11 April 2025

An optimised AI-driven swarm-based enhanced task scheduling model for cloud computing environment

Surinder Kaur* and Jaspreet Singh

Department of Computer Science and Engineering,

Chandigarh University,

Gharuan, Mohali, Punjab, 140413, India

Email: surinder.cse@cumail.in

Email: cec.jaspreet@gmail.com

*Corresponding author

Vishal Bharti

MMEC,

Maharishi Markandeshwar (Deemed to be University),

Mullana-Ambala, Haryana, India

Email: mevishalbharti@yahoo.com

Abstract: Task scheduling in cloud computing environment becomes difficult when complexity level of dispute, including task count and computing resources, rises with user growth. Solving this, an enhanced task scheduling (ETS) model with optimised artificial intelligence driven by swarm is proposed in paper. In proposed method, supervised machine learning algorithm, artificial neural networks (ANN) with swarm-based moth flame optimisation (MFO) is used to balance scheduling. MFO optimises by separating out virtual machines (VMs) considering basic properties like CPU utilisation, memory and bandwidth. ETS model is optimised based on resource allocation and balancing issues using back-propagation algorithm (BPA) with ANN (ANN-BPA) to analyse scheduling and problem identification mechanism. Efficiency of ETS model is assessed, focusing on aspects such as task allocation, task completion, execution time and energy consumption. The ANN-BPA-based task scheduling model outperforms by present technique and ANN-based model, which enhances resource utilisation by 7.54% and decreases completion time by 0.6 s.

Keywords: cloud computing; resource allocation; task scheduling; ANN-BPA; particle swarm optimisation; PSO; artificial bee colony; ABC; CSA; moth flame optimisation; MFO.

Reference to this paper should be made as follows: Kaur, S., Singh, J. and Bharti, V. (2025) 'An optimised AI-driven swarm-based enhanced task scheduling model for cloud computing environment', *Int. J. Cloud Computing*, Vol. 14, No. 1, pp.25–53.

Biographical notes: Surinder Kaur is a PhD research scholar at the Chandigarh University, Punjab, India. She has received her Master's in Computer Science Engineering. Her areas of interest are cloud computing, artificial intelligence and machine learning.

Jaspreet Singh is currently working as an Associate Professor in the Department of Computer Science and Engineering, Chandigarh University, Mohali, Punjab, India. He has published more than 80 research papers in national and international acclaimed journals and conferences. He is an active member of various professional bodies like IETE, ISTE, etc. He also published many patents and has many awards to his credit. He has more than 12 years of teaching experience and as well as guiding many postgraduate and PhD students. His major fields of interests are cloud computing, IOT, data science and network security.

Vishal Bharti is working as a Professor and the Principal at the MMEC, Maharishi Markandeshwar (Deemed to be University), Mullana. Previously, he worked as a Professor and Additional Director in the Department of CSE at Chandigarh University, Mohali, Punjab. He is having a mixed bag of experience of 17+ years in both academia and IT industry. He has seven filed patents, 28 copyrights and 13 government grants (SERB, DST, NSTEDB, EDI, etc.) and two seed money grants to his credit. He published 90+ research papers at both national and international level. His area of specialisation is cyber security, network security, cloud computing and distributed computing.

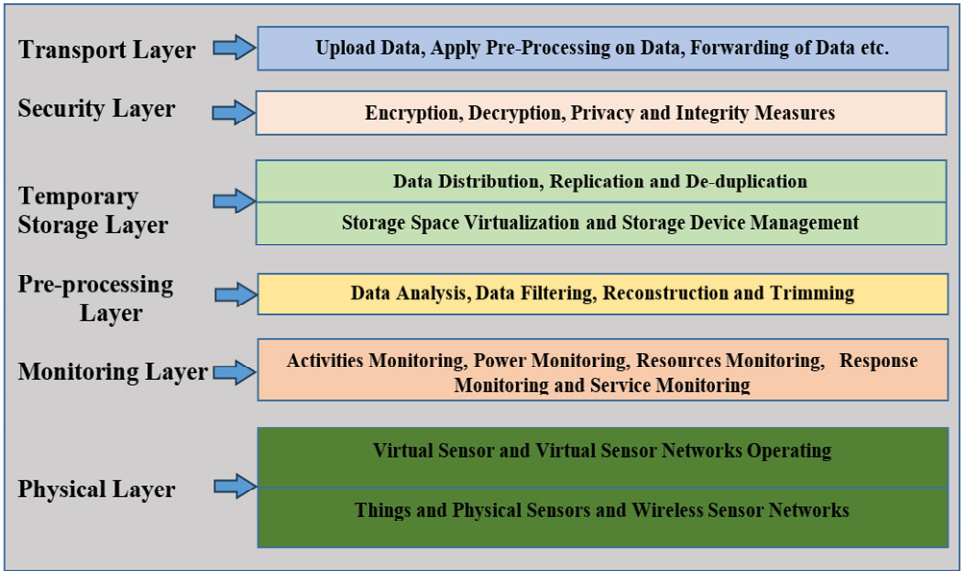
1 Introduction

Currently, there is a strong trend towards using high-performance parallel and distributed systems for various tasks due to their ability to handle large-scale computations with minimum time and efficiently. Cloud computing enables clients to access pooled computer resources over the Internet by utilising parallel and distributed computing techniques (Li et al., 2019b). The ‘consumption-based pricing’ business model has made cloud computing accessible to a wider range of users in this world. At this level of software deployment, vendors of cloud suppliers of services, and end users are engaged in performing complex tasks digitally (Singh et al., 2021). Cloud service providers provide consumers with computing capabilities using the concept of virtual machines (VMs) (Pang et al., 2019). Service providers use these VMs to provide customer services at the application level. They use task scheduling algorithms to allocate client workloads VMs, with the goals of reducing response times, ensuring a high quality of service (QoS), and optimising resource utilisation (Li et al., 2019a). Therefore, the job or task scheduling algorithm is of utmost importance in any cloud architecture that is based on a layer mechanism. Figure 1 depicts the cloud architecture.

Cloud requires modifications to the multiple scheduling approaches employed in diverse computer settings. A scheduling strategy that is optimised for a cluster may exhibit worse performance in a cloud environment (Arri and Singh, 2021). Prior to the algorithm addressing the structure of the cloud environment, it is necessary to transfer the components of the approach into the issue space. The quantity of feasible task configurations rises in proportion to the diversity of VMs and the magnitude of the workloads being handled (Krishnasamy et al., 2023). Discovering the most efficient management way among all possible arrangements is a highly complex issue in the field of cloud computing (Bindra and Aggarwal, 2024). Cloud computing involves the provision of many services, including processing power, data storage, and network access, using internet-based technology (Sreenu and Sreelatha, 2019). The idea includes

infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS), which allow enterprises to expand, foster innovation, and delegate information technology (IT) operations (Taghinezhad-Niar and Taheri, 2022). The design of a cloud-based platform primarily relies on efficient resource allocation through work scheduling in a well-balanced manner (Nematollahi et al., 2023). In this research framework, we developed a novel task scheduling with load balancing concept using an enhanced task scheduling (ETS) model with optimised artificial intelligence (AI) driven by swarm-based moth flame optimisation (MFO) to perform the task scheduling in a balanced manner. Here, the concept of artificial neural network (ANN) is used by integrating the mechanism of back-propagation algorithm (BPA) that is named as ANN-BPA. For the selection ANN-BPA as an AI technique, a comparison with other AI techniques is done. For the efficiency validation, results are compared with simple ANN as an AI technique to show the ANN-BPA proficiency for the ETS model. Both service suppliers and customers may get contentment with respect to of resource consumption and user enjoyment by using a load balancing technique and a metaheuristic strategy.

Figure 1 Working of later-based cloud architecture (see online version for colours)



Source: Arri and Singh (2021)

1.1 Our contribution

The paper introduces a novel idea of task or job scheduling incorporating load balancing in scenario of cloud. This model integrates the concept of AI along with swarm intelligence approach MFO to optimise task scheduling processes efficiently. Firstly, the proposed method performs a comparative study on several swarm intelligence algorithms, inspired by the collective behaviour of social organisms such as ants, bees, or birds, moths are utilised to optimise the process of task scheduling in cloud environments. This proposed strategy utilises the combined knowledge and distributed

decision-making skills of swarms to enhance the efficiency and efficacy of job distribution and completion in a balanced way to VMs. Some major contributions are written as:

- 1 Conduct a sort survey related to the proposed model ETS model with optimised AI-driven by swarm in cloud computing environments.
- 2 In order to enhance load balancing, it is necessary to ensure that requests are distributed in a more equitable manner, taking into account the processing capabilities of each VM. The enhanced VM load balancing led to much greater time savings compared to earlier studies.
- 3 When evaluating the compatibility of user requests with VMs, it is important to consider the whole range of resources available, including storage, RAM, and bandwidth, rather than only focusing on the CPU.
- 4 In order to satisfy the requirements of service providers and consumers, it is necessary to establish a fitness function that reduces time and optimises resource utilisation.
- 5 Implementing a cohesive approach that takes into account the needs of all service providers and customers will simplify previous approaches to task scheduling activities.
- 6 The coupling of the MFO & ANN-BPA for work scheduling along with load balancing may be efficiently achieved.
- 7 At the last, to validate the efficiency of hybrid algorithms with MFO and ANN-BPA for ETS model, a comparative analysis with other approaches is performed.

For the mentioned contributions, we have created a sort and results-based review of existing related work for task scheduling system that incorporates load balancing. Section 2 contains a comprehensive analysis of the existing works. It also includes detailed information on several AI and swarm approaches to task scheduling in cloud computing. Additionally, in Section 2, there are few gaps are highlighted that are identified by survey for various methods of scheduling tasks. Section 3 provides further information on the design ETS model with ANN-BPA along with the MFO technique. It begins by defining key terms and explaining the MFO technique. The problem is then defined and a framework called the prospered framework, which combines MFO and ANN-BPA, is discussed. This section covers the ANN-BPA model as well as a model with only ANN and MFO to show the system efficiency in comparative manner. Further, the paper's Section 4 provides a detailed analysis of the experimental configuration in the cloud ecosystem for the ETS model. The paper contains information on the results, experimental setting, and comparative examination. At last, Section 5 outlines the research's suggested conclusion on the future possibilities of work scheduling and load management in the context of cloud computing.

2 Literature survey

This research paper segment analyses numerous previous studies to identify the challenges associated with task scheduling in balanced manner for cloud computing using various swarm and AI approaches or algorithms.

Ahmed et al. in 2024 had performed a comparative analysis to find out the better approach in terms of execution time by developing a model in an efficient task scheduling manner for cloud using energy management algorithm (EMA). Researchers have used the concept of shortest job first (SJF), first come first served (FCFS) and EMAs. The experimental findings indicate that augmenting the quantity of VMs significantly decreases the Make span. The energy management algorithm (EMA) surpasses the SJF strategy by 2.79% for the cyber-shake process and excels the FCFS approach by 12.28%. EMA attains outcomes that are 21.88% superior than the total performance of both algorithms. In the Montage process, EMA provides a 4.50% enhancement compared to SJF and a 25.75% superiority over the FCFS approach. Furthermore, simulations were performed to assess the suggested mechanism in comparison to commonly used energy-efficient methods. The findings validate that the suggested structural design efficiently reduces data consumption and offers improved scheduling for cloud settings. In same year, Khan et al. (2024) proposed a model with whale optimisation algorithm (WHO) in parallel manner to enhanced the efficiency for independent tasks scheduling on the environment of cloud computing. This research introduces a parallel enhanced whale optimisation algorithm (PEWOA) to tackle the NP-hard problem of task scheduling in cloud settings. PEWOA significantly lowers make span and enhances resource usage and throughput by increasing solution diversity and circumventing local optima with a modified encircling manoeuvre and an adaptive bubble-net assaulting mechanism. A parallelisation technique guarantees minimal execution time regardless complexity, and tests conducted on CloudSim with GoCJ and HCSP datasets demonstrate its better performance compared to WOAmM and MRMPSO. The method regularly provides superior scalability and statistically significant outcomes, demonstrating its efficacy in improving cloud work scheduling. In 2023, Nematollahi et al. created a model that combines different algorithms to efficiently schedule tasks and allocate resources with the help of swarm-based mechanism. The authors proposed an optimised MFO method for efficiently scheduling tasks in a balanced manner for the internet of things (IoT) using fog computing. To enhance work scheduling is based on the swarm-based MFO, the researchers incorporate the concept of hybridisation by utilising the opposition-based learning (OBL) algorithm to facilitate resource distribution. The results suggest that the hybrid OBL-MFO model that was created resulted in a 12.18% drop in the delay factor and a 6.22% reduction in energy usage. In the field of cloud computing as well as mobile-edge computing (MEC) research was proposed by Amer et al., in 2022 for prioritisation and the management of shared resources. The study specifically emphasised prioritisation and the direction of shared resources. The researchers introduced optimised collaborative scheduling (OCS); a distinctive methodology tailored for the efficient scheduling of high-priority jobs. The OCS method operates on two separate levels. The first level involves a centralised orchestrator layer that takes on the task of scheduling jobs on MEC. In contrast, the second level, situated within the orchestrator, is accountable for assigning tasks to neighbouring cloud data centres. The study highlighted the need of assigning tasks based

on their prioritisation, while still meeting established requirements for throughput and latency. To enhance efficiency, the authors employed optimisation techniques to allocate resources in an optimal manner, ensuring that services are delivered in a cost-effective way. The experimental results, obtained from numerical calculations, have demonstrated the effectiveness of the strategy in terms of CPU utilisation rates. However, the analysis also identified areas that may be improved, specifically in reducing costs associated with offloading and decreasing average time delays. These concerns could be effectively addressed by applying various meta-heuristic techniques. In 2021, Zhu et al. proposed a strategy for scheduling work in several cloud environments that takes into account security and reliability needs. Given the growing demand for multi-cloud systems, ensuring security and dependability are paramount. The authors focus on addressing trust issues in heterogeneous multi-cloud environments and present a solution called matching and multi-round allocation (MMA) for efficient scheduling. The primary objective was to develop an MMA system that minimised expenses, optimised job completion time, and ensured security. One of the two subgroups identifies the most suitable candidate resources in terms of performance, security, and reliability. The second group aims to optimise execution time and cost by minimising the uncertainty in expected completion time through many rounds of reallocation. Swarm-based techniques such as modified CS (MCS), modified artificial bee colony (MABC), hybrid chaotic particle search (HCPS), max-min, and min-min were also employed. The novel technique decreases the amount of time required for production, lowers prices, and optimises the utilisation of resources, all while enhancing the ability to alter time and cost. In 2021, Surjeet et al. employed metaheuristics such as particle swarm optimisation (PSO) and ant colony optimisation (ACO) to achieve equilibrium between cloud computing demand and VM scheduling. This work employed two swarm-based meta-heuristic optimisation approaches to schedule VMs. The PS-ACO hybrid technique integrates the PSO and ACO algorithms. The hybrid PS-ACO algorithm utilises previous data on VM and task submissions to accurately forecast future task submissions and resource needs, without the need for further data. In addition, the proposed approach denies compute requests that surpass resource limitations. The time required for computation scheduling is decreased. The experiment demonstrates that the authors' metaheuristic algorithm effectively maintains load balance in response to environmental changes, surpassing current techniques.

In 2021, Shukri et al. conducted research on the cloud-based enhanced multi-verse optimiser (EMVO), which involves scheduling jobs depending on specific requirements. Task scheduling is a challenging process in cloud computing, since it requires minimising execution time and cost while efficiently utilising resources. In the past decade, numerous academics have utilised swarm-based meta-heuristics such as multi-verse optimiser (MVO) and PSO for scheduling cloud work. The authors of this paper developed EMVO, an enhanced MVO task scheduler. Based on the findings, the authors' EMVO algorithm demonstrates superior performance compared to MVO and PSO algorithms in cloud environments. According to a study, EMVO performs better than MVO and PSO in terms of make span time and resource utilisation. Mohammed (2021) is conducting a comprehensive investigation of bio-inspired and integrity check procedures for storing data across many cloud platforms in 2021. This study proposed an innovative data integrity solution for cloud computing clients that have service providers with lower reliability. This approach yields reliable metadata and operates more efficiently than previous techniques, resulting in cost savings for consumers. In 2021, Pirozmand et al. employed a natural computing-based hybrid genetic algorithm (HGA) with

multi-objective optimisation to address the cloud work scheduling problem. HGA, or HGA, is a method that involves two distinct steps to efficiently schedule processes that have constraints related to both energy and time. HGA initially establishes the priority of each task and subsequently allocates the tasks to processors. They first organised their workloads based on priority and generated HGA main chromosomes. Then, they allocated processor tasks using the energy-conscious scheduling heuristic (ECSH) paradigm. The simulation results indicate that the suggested HGA plus ECSH model performs better than other options. In 2021, Mubeen et al. devised an adaptive load balancing strategy for scheduling tasks in cloud computing. The study indicates that ALTS is the strategy employed for adaptive load balancing in task scheduling. The author's survey indicates that several mechanisms such as GA, PSO, ACO, and others have been proposed for task scheduling. However, there is a need for improvement in some scheduling parameters. The authors introduced a load-balanced ALTS technique for scheduling cloud computing tasks. The goal is to reduce the make span by efficiently utilising available VMs and minimising SLA violations. The authors confirm the effectiveness of the recommended strategy by comparing it to existing works that rely on ARU, make span, and SLA violation. They conclude that their approach outperforms GA, ACO, and GA2CO. According to the survey provided, there are some significant limitations in the current models, which are described below:

- 1 The majority of swarm-based algorithms operate through iterative processes, in which a population of solutions evolves over time, resulting in energy consumption and wastages of resources (Shukri et al., 2021).
- 2 One major disadvantage is their vulnerability to convergence problems of most of the swarm-based approaches. When algorithms are not calibrated correctly or when the optimisation landscape is very complicated, they may converge too early to suboptimal solutions, hence failing to effectively explore the search space (Zhu et al., 2021; Mohammed, 2021; Pirozmand et al., 2021).
- 3 Frequently utilised techniques typically require significant processing resources. As the quantity of tasks and limitations in scheduling difficulties grows, the computational expense can become excessively costly (Mubeen et al., 2021). As a result, they are not well-suited for scheduling applications that require real-time or large-scale capabilities.

So, to solve such kind of problem, the paper proposes a model having foundation MFO along with ANN-BPA based on some comparative analysis in the next section of paper. Our solution decreases the duration needed to complete a make-span, while improving resource utilisation in balanced way and the results of the existing model summary is demonstrated in Table 1.

Upon reviewing previous research related to the topic of task scheduling, the methodology section plays a crucial role in the research study by outlining the systematic strategy used to address the research objectives. The literature review establishes a strong basis, while the methodology section details the precise stages, procedures, and techniques used to gather, analyse, and interpret data relevant to the task scheduling with load balancing concept in single as well as multi cloud environment. The used method and material of the proposed ETS model with optimised AI-driven by swarm in cloud computing environments in explained in the below section of paper.

Table 1 Existing model summary

<i>References</i>	<i>Applied method</i>	<i>Approach</i>	<i>Advantage</i>	<i>Drawback</i>
Nematollahi et al. (2023)	• Moth flame optimiser (MFO) • Osmosis load balancing (OLB)	Task scheduling and allocate resources with the help of swarm-based mechanism	• Minimising make-span of VMs • Minimising the time required to complete a job	• Model is not scalable • The metrics' import level is not used.
Amer et al. (2022)	• Optimised collaborative scheduling (OCS)	Prioritisation and the management-based resources sharing for task scheduling	• Faster processing time • Operative resource consumption	• Assign non-redemptive dependent tasks to the VMs according to a schedule.
Zhu et al. (2021)	• Managed cloud services (MCS) • Modified artificial bee colony (MABC) • Hybrid chaotic particle search (HCPS) • max-min • min-min	Task scheduling work with security and reliability needs	• Cost reduction in processing • Optimisation of transit duration • Viability in diverse systems	• Lack of scalability is evident. • The significance level of the measures is not used.
Surjeet et al. (2021)	• Particle swarm optimisation (PSO) • Ant colony optimisation (ACO) • Hybridisation of particle swarm optimisation (PSO) and ACO	Work on to achieve equilibrium between cloud computing demand and VM scheduling	• Minimum job execution time • Limited task transfer time	• Execution of tasks based on mutual independence
Shukri et al. (2021)	• Enhanced multi verse optimiser (EMVO) • Multi verse optimiser (MVO) • PSO	Task scheduling depending on specific requirements	• Duration of task transfer Costs of execution • Optimal energy efficiency • Minimum duration of task sequence	• Significant amount of time dedicated on job scheduling and slow pace
Mohammed (2021)	• Bio-encouraged method • Integrity proof mechanism	Storage of information protection in a multi-cloud context	• Minimum costs	• Scalability is not available and achieving balance is challenging.

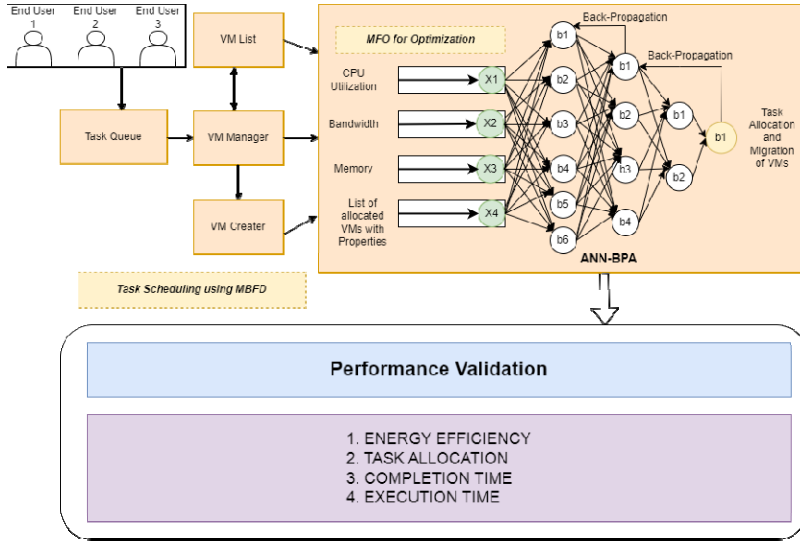
Table 1 Existing model summary (continued)

<i>References</i>	<i>Applied method</i>	<i>Approach</i>	<i>Advantage</i>	<i>Drawback</i>
Pirozmand et al. (2021)	• Hybrid genetic algorithm (GA)	Multi-objective optimisation to address the cloud work scheduling problem	• Low execution costs • Less energy	• Scalability is not provided and balancing is difficult
Mubeen et al. (2021)	• Genetic algorithm (GA) • PSO • ACO • GA2CO	An algorithmic technique for effectively distributing computational tasks in cloud to optimise performance and resource use.	• Better ARU and make span • Minimum SLA violation	• Transfer of interdependent duties • Inadequate degree of scalability
Ahmad and Khan (2020)	• PSO • Resource aware least busy (RALB) • Honey bee foraging	An hybridised algorithm developed from PSO and RALB inspired from honey bee behaviour to optimise task scheduling	• Minimised makespan time • Less processing time • Decreased processing cost	• Non availability of scalability

3 Material and method

This research paper provides an overview of the working architecture of proposed ETS model that focuses on comparing different approaches to task scheduling in cloud computing environments. The study specifically explores the use of AI techniques with swarm-based MFO approach for task scheduling optimisation purposes. For the selection of MFO as a swarm approach for the proposed ETS model, this study conducts a comparative survey on various swarm-based approaches like PSO, artificial bee colony (ABC) and CSA with MFO and founded that the performance of MFO is far better than other with ANN-BPA as an AI approach (Kaur et al., 2024). But, for the selection of ANN-BPA, a comparative analysis is performed with simple ANN mechanism in this research. Basically, projection of an ETS model by utilising the concept of MFO along with ANN-BPA is made and to validate the efficiency of the model, comparison is done with various existing works. The developed model for the simulation is shown in the Figure 2. This ETS model works for the all-proposed combination of algorithms such as ANN and ANN-BPA with MFO as a swarm-based optimisation technique for the task scheduling in cloud computing.

In this research, the concept of ANN-BPA is used along with the MFO to develop an ETS model to perform the task scheduling in a balanced manner to optimise the performance of the model in terms of task allocation, completion, execution time, and energy consumption.

Figure 2 Proposed ANN-BPA-based architecture of ETS model (see online version for colours)

3.1 Using MBFD algorithm to schedule tasks

The modified best fit decreasing (MBFD) algorithm is a well-known task scheduling method employed in cloud computing settings to effectively distribute resources and handle activities according to their deadlines and resource demands. The MBFD algorithm arranges tasks based on their resource requirements, giving priority to those with greater resource demands, and sorts them in descending order. Moreover, task that have closer deadlines are prioritised above tasks that have later deadlines. The method sequentially chooses the task with the greatest resource requirement and earliest deadline from the other tasks, and assigns it to an accessible cloud resource. This process continues until all tasks have been scheduled or until there are insufficient resources to fulfil the requirements of the remaining tasks. The algorithm of MBFD is written as:

Algorithm 1 MBFD for task scheduling

I/P: $H_{LIST} \rightarrow$ List of Host

$VM_{LIST} \rightarrow$ List of VMs (50 to 100VMs)

O/P: $VM_{SORTED} \rightarrow$ List of sorted VMs

1 Start algorithm

- 2 Perform a sorting operation on VM_{LIST} based on their CPU utilisation,
Memory,
Bandwidth.

- 3 Find the size of host list, $Count_H = \text{Size}(H_{LIST})$

- 4 Find the size of VM list, $Count_{VMs} = \text{Size}(VM_{LIST})$

5 For each $Count_H$

- 6 Determine the power of VMs, $Minpower \rightarrow \max$

- 7 Generate an empty matrix to contain the sorted values of VMs, $VM \rightarrow [\text{null}]$

```

8      For each CountVMs
9          Power allocation, VMPOWER → Power assessment (VMLIST(VMs), HLIST)
10     If VMPOWER is lower than average VMPOWER, then
11         Allocation is performed for VM to List → VMLIST(VMs)
12         Assign power as Min (VMPOWER) → VMPOWER
13     If Allocated HLIST is not null, then
14         Allocation of VM is performed
15         VMSORTED → Sorting is performed for VMs based on the VMPOWER
16         End – 1 If
17     End – 2 If
18 End – 1 For
19 End – 2 For
20 Return: VMSORTED list of VMs sorted in ascending order
21 End – MBFD Algorithm

```

Firstly, this research performs the task scheduling using the traditional algorithm that is MBFD (Modh et al., 2022) and the flowchart of MBFD algorithm is shown in the Figure 3.

As a starting, firstly initialise some basic parameters such as Number of Tasks, cloud servers (CSs), VMs. That is clearly mentioned in the Figure 4. After that all-initial parameters are sorted for the allocation purpose.

When the task scheduling for proposed ETS model is performed, then this study use the concept of swarm-based MFO for the optimisation of resource allocation in balanced manner according to the fitness function.

3.2 MFO to optimise tasks scheduling in load balanced manner

When the task scheduling is performed by MBFD algorithm then the concept of balancing of load is not considered and due such scenarios, VMs will be overloaded/underloaded. To short out such kind of problem, the concept of MBFD is used in the proposed ETS model with a novel fitness function. The MFO algorithm is a metaheuristic optimisation technique that emulates the behaviour of moths in nature to attracted towards the light. The concept relies on the innate tendency of moths to be attracted to a source of light, which offers a possible resolution to an optimisation problem related to the task scheduling in cloud. The MFO algorithm can be utilised to effectively allocate computational task across VMs in a cloud computing, precisely in the context of load balancing in task scheduling. This is the operational process:

- 1 *Initialisation of moth:* The MFO algorithm commences by creating an initial population of moth according to the potential solutions. Each solution represents a latent assignment of tasks to VMs. In the scenario of task scheduling, initialised populations are represented as matrices, and a viable method involves distributing tasks across heterogeneous VMs using a $m \times n$ position allocation matrix. In order to demonstrate the most effective approach for allocating tasks among diverse VMs, can employ the terms ‘particular best (pBest)’ and ‘global best (gBest)’, which represent matrices consisting of 0s and 1s, singly.

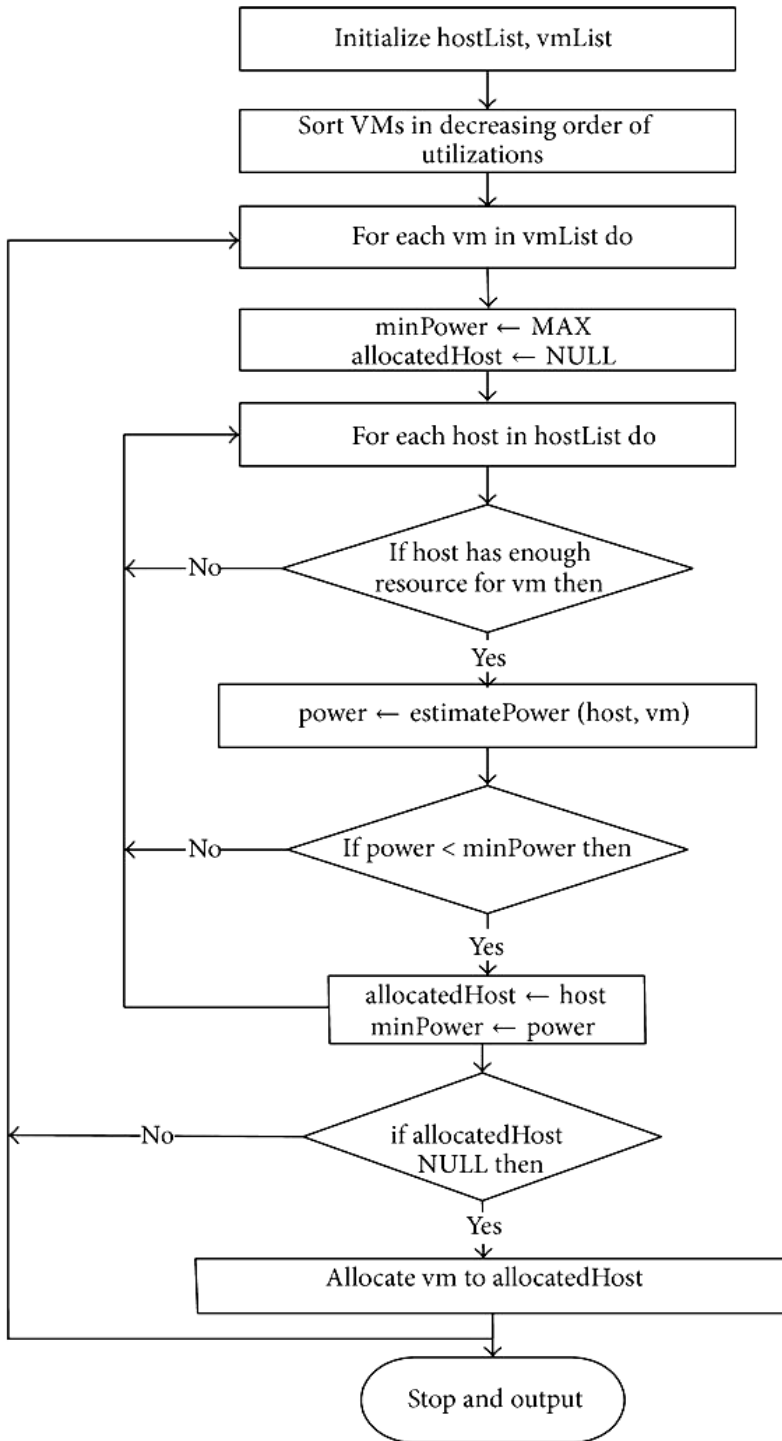
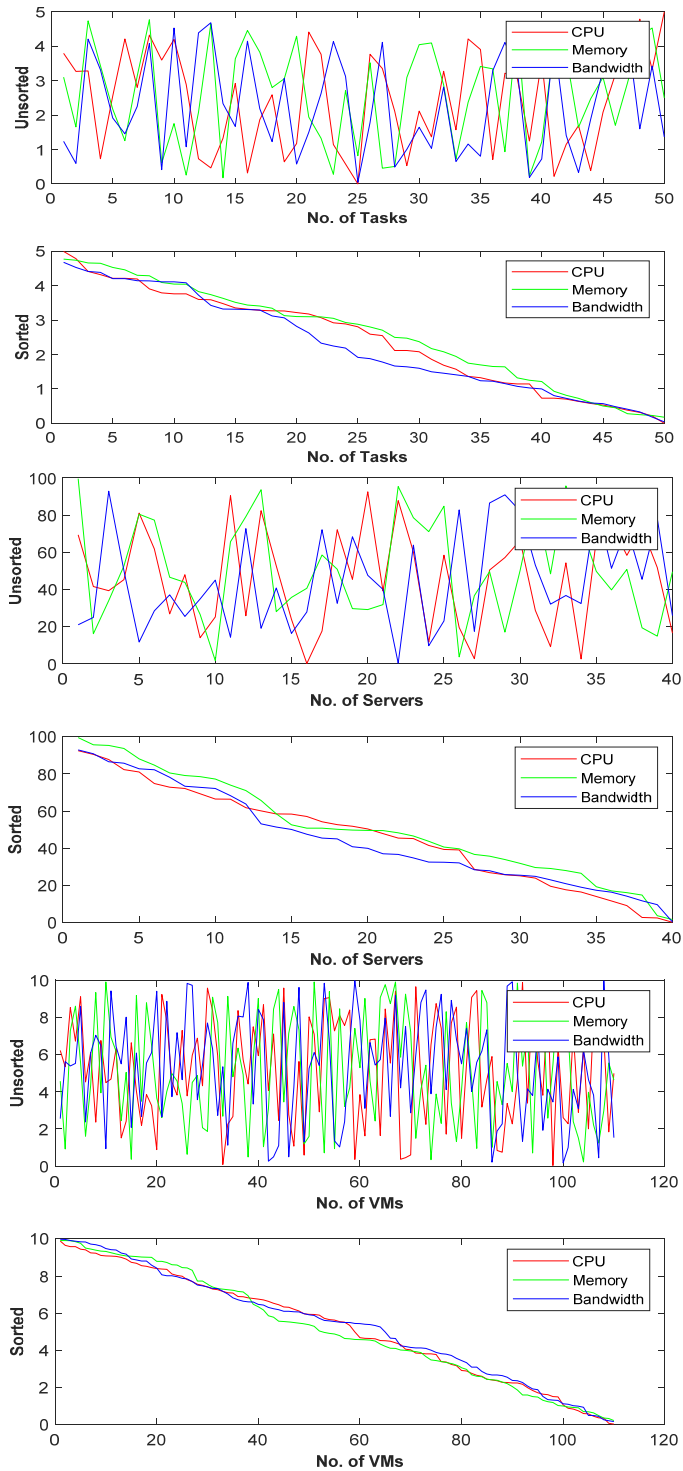
Figure 3 Flowchart of MBFD for proposed ETS model

Figure 4 Initial parameters of the model for simulation in cloud (see online version for colours)

- 2 *Calculation of fitness:* The fitness of any solution is evaluated according to predetermined criteria, such as available resources (CPU utilisation, memory and bandwidth), minimising the total execution time, energy consumption, or maximising resource use. The used fitness function of MFO is written as:

$$fit_{fun} = \begin{cases} 1; & \text{if VM have allocation power } (C_p > T_p) \\ 0; & \text{otherwise} \end{cases} \quad (1)$$

where C_p is the random change in moth and T_p is the threshold which should be achieved by moth

- 3 *Light source attraction:* During each iteration of the MFO algorithm, moths (symbolising various solutions) are drawn towards the light source (symbolising an optimal solution). This movement is driven by two primary forces:
- A behaviour in which moths are drawn towards a light source and this behaviour can be interpreted as their inclination towards finding more optimal solutions within their location.
 - Moths engage in stochastic motions to navigate the search area and prevent themselves from becoming stuck in local optimal solutions during the allocation of task.
- 4 *Revised solutions:* Solutions are adjusted according to the updated positions of moths with the help of pBest and gBest as they are attracted to the light source. This process continues in an iterative manner until a stopping requirement is satisfied, such as reaching a predetermined maximum number of iterations or attaining a solution that is deemed satisfactory for the allocation of VMs.
- 5 *Solution selection:* Ultimately, the most favourable solution achieved once the algorithm reaches a stable state is chosen as the ideal distribution of tasks to VMs. This ensures that the burden is distributed fairly and that resources are used efficiently in the context of the cloud. This phase will facilitate the creation of two distinct categories: underloaded/overloaded and normal. The procedure for working with the MFO algorithm is as follows:

Algorithm 2 MFO for task scheduling optimisation

I/P: $H_{LIST} \rightarrow$ list of host

$VM_{LIST} \rightarrow$ list of VMs

$VM_{SORTED} \rightarrow$ list of sorted VMs

O/P: $VM_{ALLOCATED} \rightarrow$ List of allocated VMs

- 1 Start
- 2 Firstly, apply MBFD, then use MFO
- 3 $M_pos \rightarrow$ Randomly assign Moth position
- 4 **For each** I **in** M_pos
- 5 Calculate the fitness function f_i
- 6 **End – For**
- 7 **While** ($itr \leq itr_max$)
- 8 Update the position of M_pos

9 Compute number of flames using given equation (2)

10 $Number\ of\ (M_F) = round\left(N^{max} - t * \left(\frac{N^{max} - t}{T}\right)\right)$ (2)

11 Where, maximum number of flames are represented by N^{max} with current iteration represented by t and the T represents the maximum number of iterations. The algorithm's stages are summarised below.

12 Evaluate fitness function (fi)

13 **If itr==1**

14 M_F= sort (M)

15 M_FO=sort (M_o)

16 **Else**

17 M_F=sort (M_(t-1), M(t))

18 M_FO=sort (M_(Ot-1), M(Ot))

19 **End – if**

20 VMALLOCATED = VMallocation(M_FO)

21 **Return:** VMALLOCATED list of Allocated VMs

22 **End – MFO Algorithm**

At last, the study used the concept of ANN and ANN with concept of back-propagation algorithm named as (ANN-BPA) for the ETS model training with MFO algorithm. The algorithm of ANN-BPA is written as:

Algorithm 3 ETS Model with ANN-BPA

Input: VMALLOCATED → List of allocated VMs

Output: ETS-Net → Trained model architecture

1 **Start**

2 Size of VMALLOCATED, [R, C] = size (VMALLOCATED)

3 Extracting labels from VMALLOCATED

4 All-Label = []

5 **For each Index in range (VMALLOCATED (R))**

6 Label (Index) = VMALLOCATED (I, Col)

7 **If Label (Index) not belongs to Label**

8 All-Label (Count) = Label (Index)

9 **End – 1 If**

10 **End – 2 For**

11 Create target for model training

12 Target class = []

13 **For each index in range (all label, row)**

14 **If ε 1st Exclusive Label**

15 Target class (1, Index) = 1

16 **Else if ε 2nd exclusive label**

17 Target class (2, Index) = 2

```

18      End – 2 If
19  End – 2 For
20  Call ANN with back-propagation algorithm with 10 hidden layers
21  ETS-Net = newff (VMALLOCATED, Target, 10)
22  ETS-Net = train (ETS-Net, VMALLOCATED, Target)
23  Test the proposed ETS model using ETS-Net
24  Result = sim (ETS-Net, test data)
25  CP = calculate parameters (result, unique label)
26  Return: CP as classification parameters
27  End – ETS model algorithm

```

Based on the above-mentioned algorithms of ANN-BPA, ETS model is simulated, and to validate the efficiency, the study compare it with a simple ANN architecture-based model along with existing works. To train the ETS model, the methods create simulation-based data for under/overloaded and normal VMs based on the fitness condition using MFO algorithm and pass to ANN-BPA or simple ANN. The simulation results based on the proposed scenarios is briefly explained in the next section of paper.

4 Results and discussion

This section of the paper presents the empirical findings obtained from our thorough investigation in the preceding sections for the proposed ETS model for task scheduling in single as well as multi cloud environments. This part provides a comprehensive analysis of the simulation of ETS model along with simple ANN as well as exiting works. In this section, firstly the paper discuss the results obtained from the simulation of ETS model to the impact the optimisation technique in terms of quality-of-service (QoS) parameters such as task allocation, task completion time, execution time as well as energy consumption in Figure 5 to Figure 8. The used simulation environment or set up of the proposed model is given in Table 2.

Table 2 Model simulation environment setup

<i>Parameters</i>	<i>Values</i>
Software	MATLAB 2020a
Toolboxes	Data acquisition, communication, optimisation and neural network
Network dimensions	1,000 × 1,000 m
No. of iterations	10–100
Optimisation algorithm	MFO algorithm
Classification algorithm	ANN-BPA
Input layers	40
Hidden layers	10
Output layers	40
Epochs	100
Total number of simulations	10

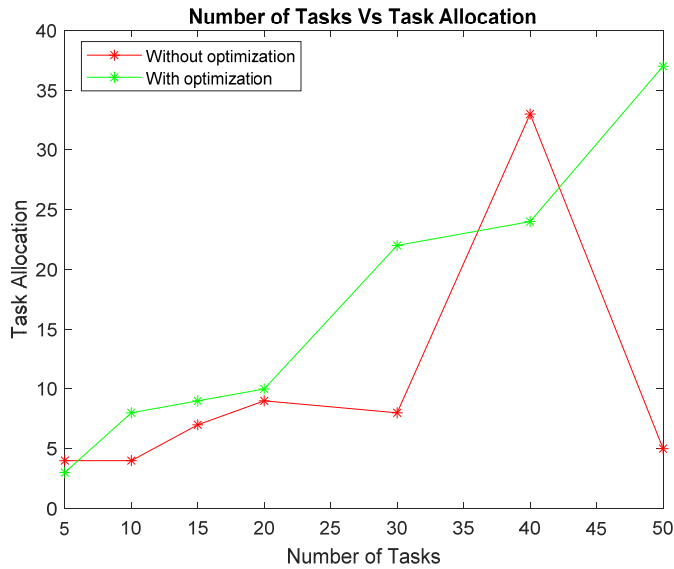
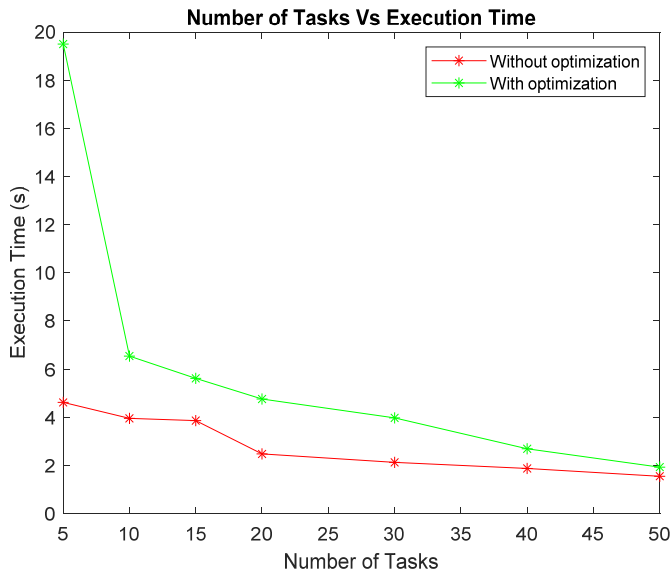
Figure 5 Task allocation execution parameter of ETS model (see online version for colours)**Figure 6** Task completion time parameter of ETS model (see online version for colours)

Figure 5 represents the obtained execution parameters of proposed model in terms of task allocation with respect to the No. of tasks in range of 5 to 50 with interval of 5. Basically, the concept of task allocation in cloud computing involves the assignment of jobs or workloads to VMs in a cloud environment. The allocation of resources is often performed dynamically, taking into account factors such as resource availability, workload characteristics, and performance needs. According to Figure 5, it is proved that the task

allocation of proposed model with MFO is gradually increases and better than the task allocation without optimisation with respect to the No. of Tasks. Where, Figure 6 represent the task completion time which are allocated to the particular VMs in cloud. In terms of task completion time, the proposed ETS model with MFO is better and complete the task with minimum time compare to the without optimisation scenarios. That means, task allocation and their execution is superior with optimisation for the ETS model.

Figure 7 Execution time parameter of ETS model (see online version for colours)

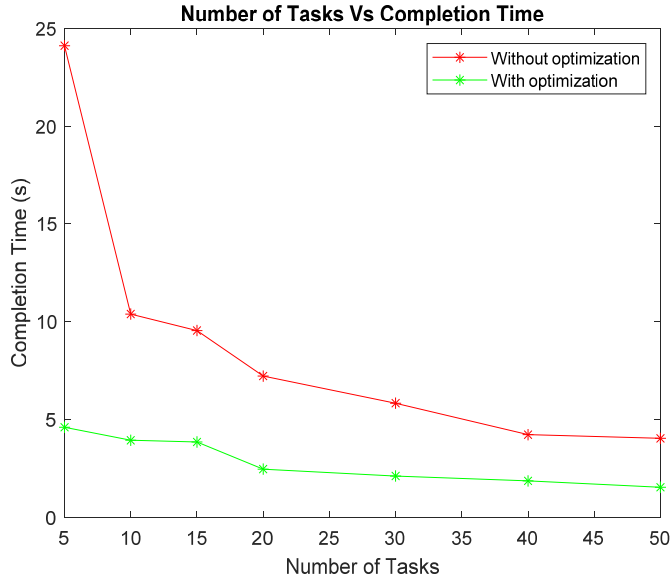


Figure 8 Energy consumption parameter of ETS model (see online version for colours)

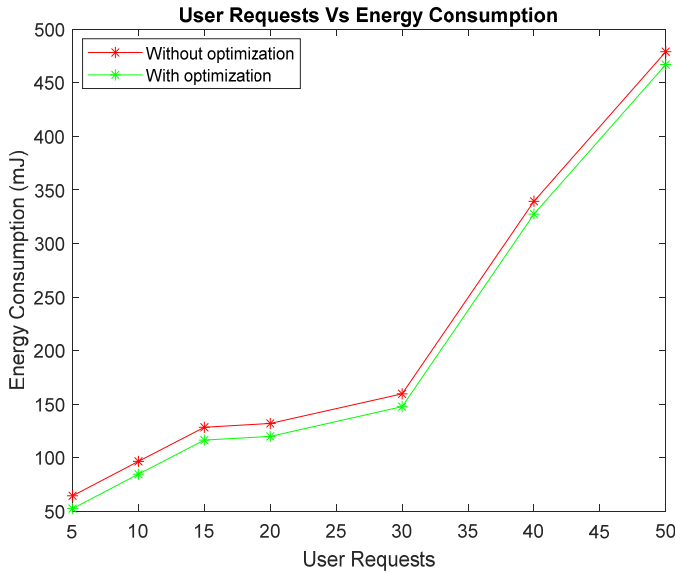


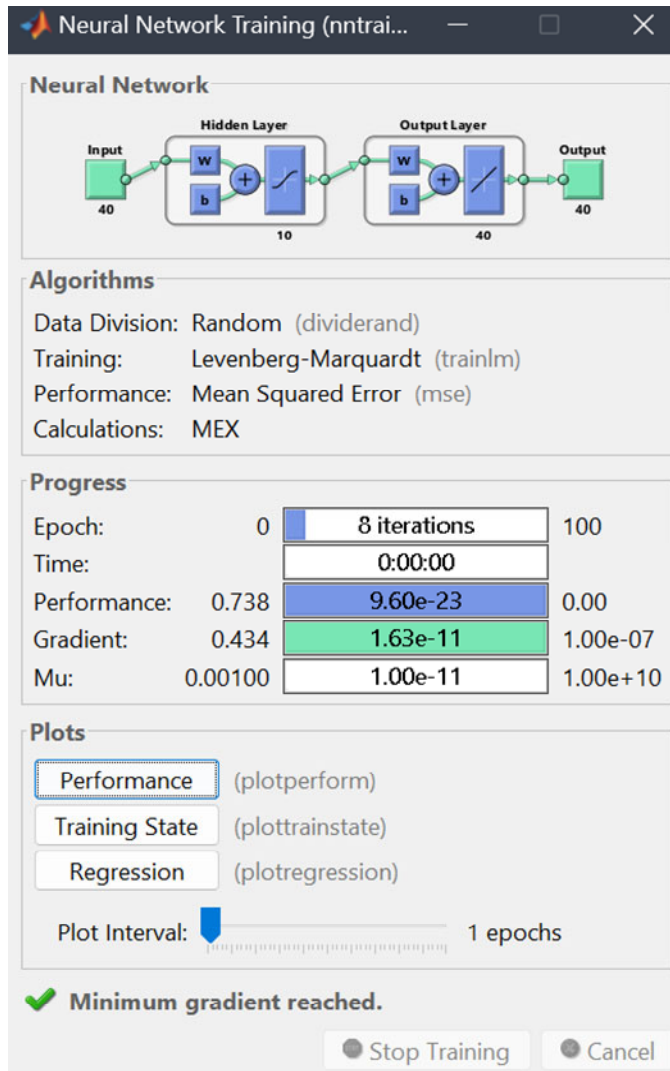
Figure 9 ANN-BPA architecture for ETS model (see online version for colours)

Figure 7 represents the execution time of proposed ETS model with respect to the No. of tasks in range of 5 to 50 with interval of 5. Execution time in cloud computing refers to the timeframe required to perform a task or workload on VMs within the cloud environment. The allocation of jobs to VMs is a critical factor in influencing the time it takes for tasks to be executed in the cloud. The execution time in cloud task scheduling for VMs is influenced by multiple factors:

- 1 The execution time of a task is greatly influenced by its task characteristics, including factors like as computational complexity, memory needs, and input/output activities. Tasks that need more processing resources may have a longer execution time.

- 2 The performance of VMs, which includes factors such as CPU speed, memory capacity, and network bandwidth, directly affects the time it takes to complete tasks. Allocating jobs to VMs that have sufficient resources can effectively decrease the time required for execution.
- 3 The task scheduling algorithm is responsible for assigning tasks to VMs depending on parameters such as job deadlines, resource availability, and load balancing requirements. Efficient scheduling algorithms aim to minimise task waiting times and resource contention, hence optimising execution time.
- 4 Load balancing is the practice of evenly distributing tasks across numerous VMs to prevent single VMs from being overloaded and to optimise the use of resources. Efficiently distributing the workload can decrease the time it takes to complete a task by avoiding bottlenecks and conflicts over resources.

From the observation, execution time of proposed ETS model with MFO optimisation technique is far better to model without optimisation approach.

Figure 8 represents the rate of energy consumption by ETS model with respect to the user requests. Energy consumption in cloud computing refers to the electrical power used by the physical servers and infrastructure components that carry out functions and manage VMs in the cloud environment. The allocation of tasks to VMs has a substantial impact on energy consumption inside the cloud computing environment. It is critical to reduce the amount of energy that servers use in data centres. It may be estimated as the total energy used during a given time period and the equation is written as:

$$EC(t) = \int_{t=0}^{send} E(t) \quad (3)$$

where E is the consumption of energy and t is the time period.

To obtain the above-mentioned parameters, the concept of ANN-BPA as a machine learning algorithm is used and the architecture of used ANN-BPA is shown in Figure 9.

Figure 9 represents the architecture of ANN-BPA in MATLAB software from the neural network toolbox (NNT). It depicts a training process for used ANN-BPA and here's a breakdown of the key elements:

4.1 Layer of ANN-BPA

- *Input layer*: This layer is responsible to take input from the ETS model based on MFO-based optimisation for each task.
- *Hidden layer*: This layer is where the main computations and learning occur within the ANN-BPA for ETS model. The number of neurons in this layer can significantly impact the network's performance and ten hidden layers based on the tuning is used.
- *Output layer*: This layer represents the final output of the ANN-BPA for ETS model. In this case, it likely has one neuron since it seems to be a binary classification problem (classifying as under/overloaded and normal VMs during the task execution in cloud computing environment).

4.2 Training parameter of ANN-BPA

- *Epoch*: This refers to a complete cycle of passing the entire training data through the ANN-BPA for ETS model once. The epoch value in Figure 9 indicates that the training process.
- *Iterations*: This represents the number of times the network has processed a single data point or a mini-batch of data points within the current epoch.
- *Performance*: This section shows the current performance metric of the ANN-BPA for ETS model during training. In this case, it displays '0.738' (the specific metric is not visible in the image, but it could be accuracy, mean squared error, or another relevant measure).
- *Gradient*: This section likely indicates the magnitude of the gradients calculated during backpropagation. Backpropagation is an algorithm used to adjust the weights of the network based on the errors between the network's predictions and the desired outputs. Here, the value is '0.434'.
- *Mu*: This might represent a learning rate parameter that controls how much the weights of the network are adjusted during training. The current value is '0.00100'.

4.3 Plots of ANN-BPA

Figure 9 shows options for different plots that can be visualised during training. These plots can help monitor the network's learning progress and identify potential issues. Some common plots include:

- *Performance plot*: Tracks the performance metric (e.g., accuracy) over epochs that is shown in the Figure 10.
- *Training state plot*: Visualises the network's outputs compared to the desired targets that is shown in the Figure 11.
- *Regression plot*: Might be relevant for regression problems, where the network predicts continuous values that is shown in the Figure 12.

Overall, the figure from Figure 10 to Figure 12 depicts the initial stage of training a neural network using the MATLAB NNT. The information displayed allows you to monitor the network's performance, gradients, and training progress that is clearly mentioned in the below figures with 8 epochs.

Based on the above scenarios presented, develop and perform the model simulation and the simulated model is shown in the Figure 13 with graphical user interface (GUI) in MATLAB. GUIs offer user-friendly interfaces that incorporate visual components like buttons, sliders, and dropdown menus to help the simulation of ETS model in cloud environment. GUIs also facilitate the access and utilisation of the ETS model for users with different degrees of technical proficiency. This level of accessibility fosters cooperation among diverse teams and simplifies the acceptance of users from various fields. This simplifies user interaction with the model, eliminating the need for advanced programming skills. In the simulation, 50 users, labelled as U1 to U50, were involved. Tasks were assigned to a specific number of CSs. Every user task is graphically indicated

by a specific colour, and the assignment of tasks to CS is shown by circles that have the same colour as the relevant user.

Figure 10 Performance of ANN-BPA for ETS model (see online version for colours)

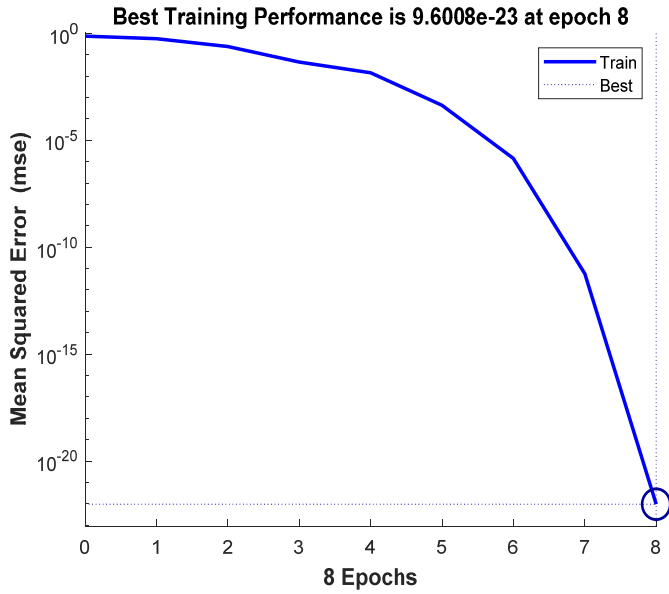


Figure 11 Training state of ANN-BPA for ETS model (see online version for colours)

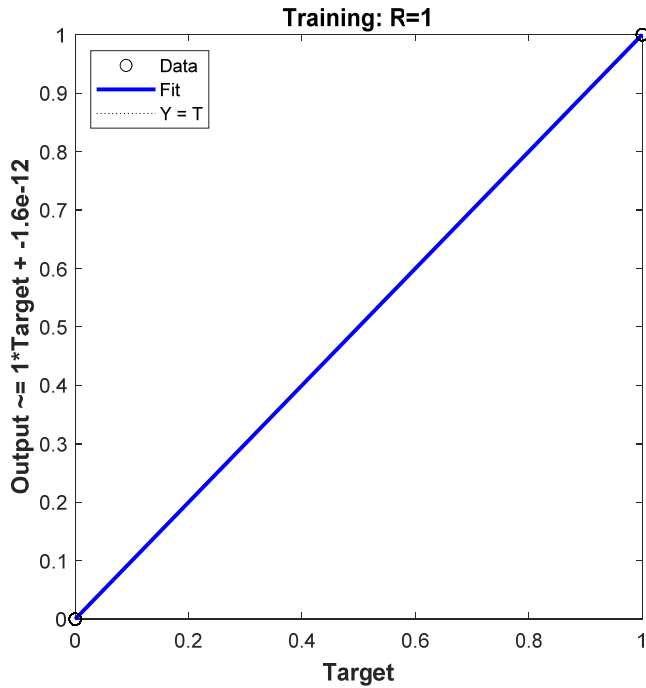


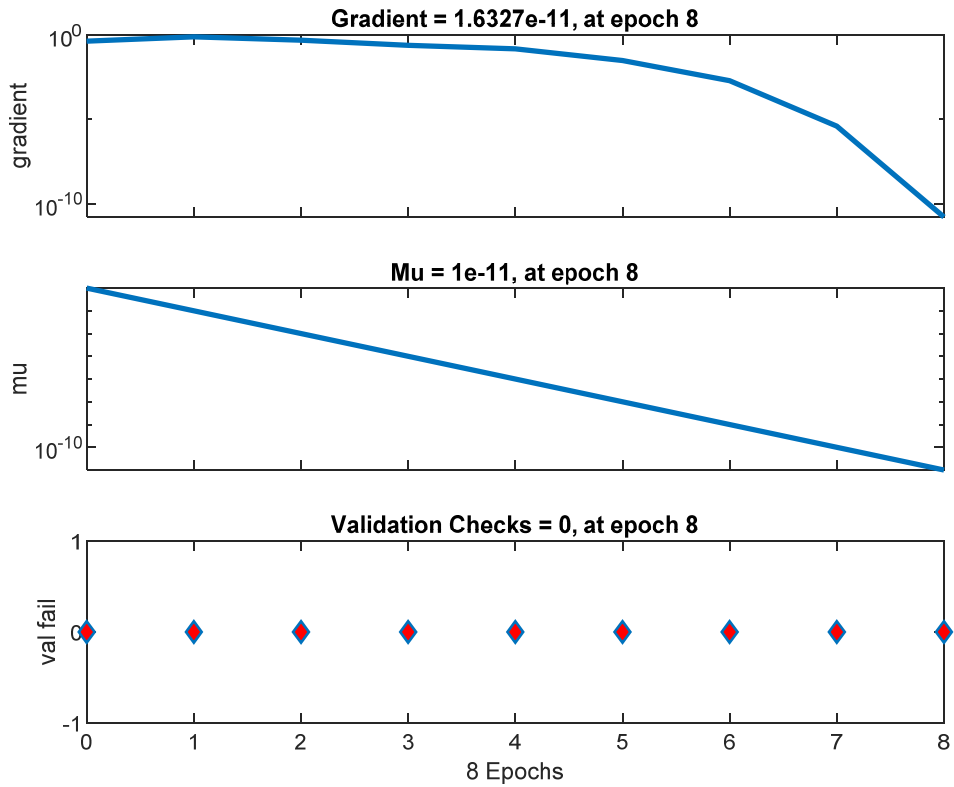
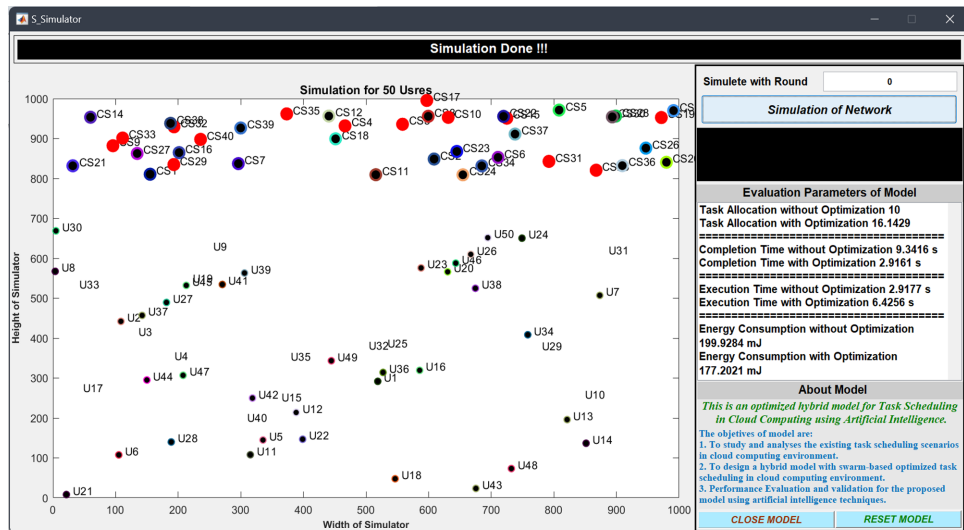
Figure 12 Regression of ANN-BPA for ETS model (see online version for colours)**Figure 13** Simulated results of developed model (see online version for colours)

Table 3 Model efficiency comparison based on energy consumption

<i>Simulations</i>	<i>Existing model (Zhu et al. 2021)</i>	<i>ETS with ANN</i>	<i>ETS with ANN-BPA</i>
10	349	301	252
20	366	317	290
30	394	374	308
40	411	391	366
50	416	397	381
60	445	420	413
70	516	497	456
80	535	516	499
90	629	573	524
100	658	584	535

Figure 14 Comparison based on the energy efficiency (see online version for colours)

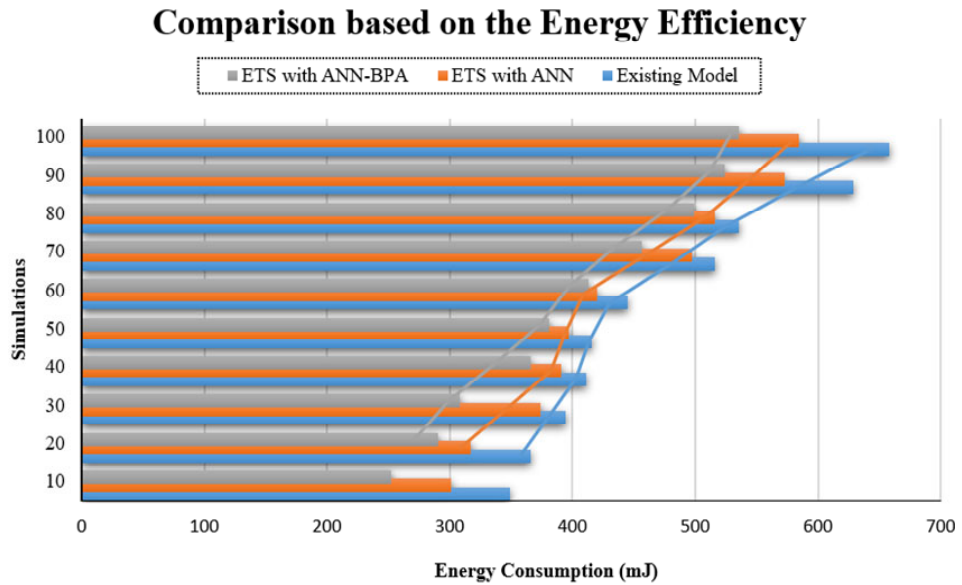
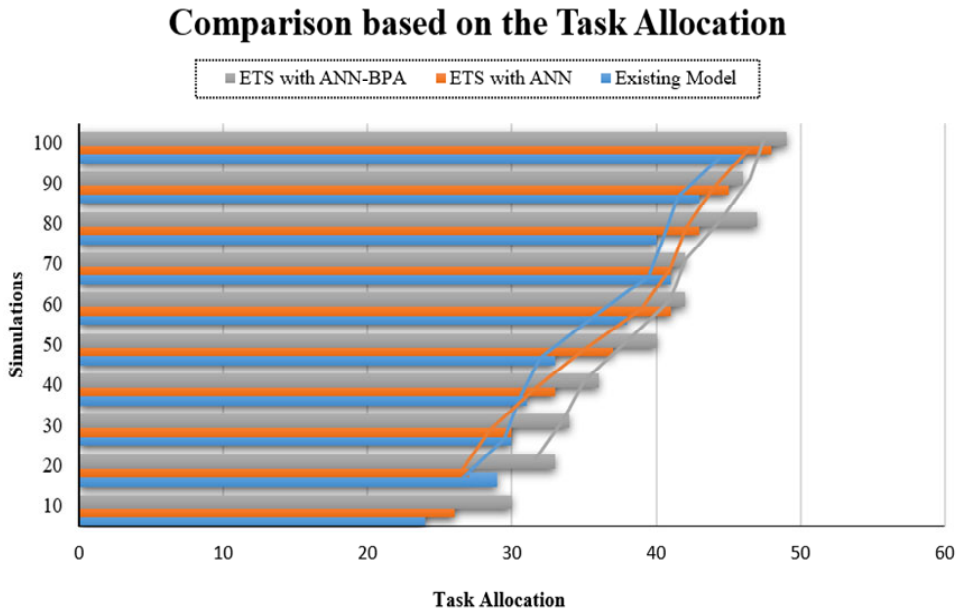


Table 4 Comparison based on the task allocation

<i>Simulations</i>	<i>Existing model (Zhu et al. 2021)</i>	<i>ETS with ANN</i>	<i>ETS with ANN-BPA</i>
10	24	26	30
20	29	27	33
30	30	30	34
40	31	33	36
50	33	37	40
60	38	41	42
70	41	41	42
80	40	43	47
90	43	45	46
100	46	48	49

Figure 15 Comparison based on the task allocation (see online version for colours)**Table 5** Comparison based on the completion time

<i>Simulations</i>	<i>Existing model (Zhu et al. 2021)</i>	<i>ETS with ANN</i>	<i>ETS with ANN-BPA</i>
10	0.89	0.76	0.55
20	0.93	0.86	0.67
30	0.99	0.93	0.73
40	1.05	0.94	0.81
50	1.6	0.98	0.88
60	1.8	1.3	0.93
70	2.3	2.1	0.98
80	2.8	2.4	1.11
90	2.9	2.88	1.71
100	3.39	3.21	2.09

In this depiction, a black circle with the user's chosen colour as its outline indicates that the CS is now occupied with executing the task provided to the particular user. This engagement may entail either using the same physical machine or assigning a VM for the task. In contrast, circles that are completely filled with the colour red indicate servers that have seen excessive load. An overloaded server is a server that is excessively burdened and unable to effectively manage the given tasks due to exceeding its capacity. The servers that are not red will process the request since they are not overwhelmed. The overloaded servers can either wait for the next request or establish a new VM to handle the user requests. This is achieved by a mix of a swarm-based method with ANN-BPA. Using the simulation, we calculated some parameters. The simulation screenshot illustrates a job scheduling system for 50 users distributed across many servers shown as

circles. The color-coded servers denote their status: black circles with colourful edges represent servers actively processing user tasks, whilst red-filled circles indicate overcrowded servers incapable of managing more requests efficiently. The performance measurements shown in the right panel encapsulate the findings, emphasising the substantial improvement attained by optimisation. Task allocation optimisation improved from 10 to 16.1429, while energy usage decreased from 199.9284 units to 177.7201 units, illustrating the system’s efficacy in resource use. Furthermore, completion and execution times were improved, resulting in superior task scheduling relative to the unoptimised system. This image clearly demonstrates how optimisation strategies improve speed by balancing server loads and reducing bottlenecks. Study will now go to the experimental portion to analyse the combined outcomes of the several planned models. This section discusses the simulation results of the swarm-based task scheduling model using ANN-BPA and also compared with simple ANN-based model and the obtained results are given in the Table 3, 4, 5 and 6.

Figure 16 Comparison based on the completion time (see online version for colours)

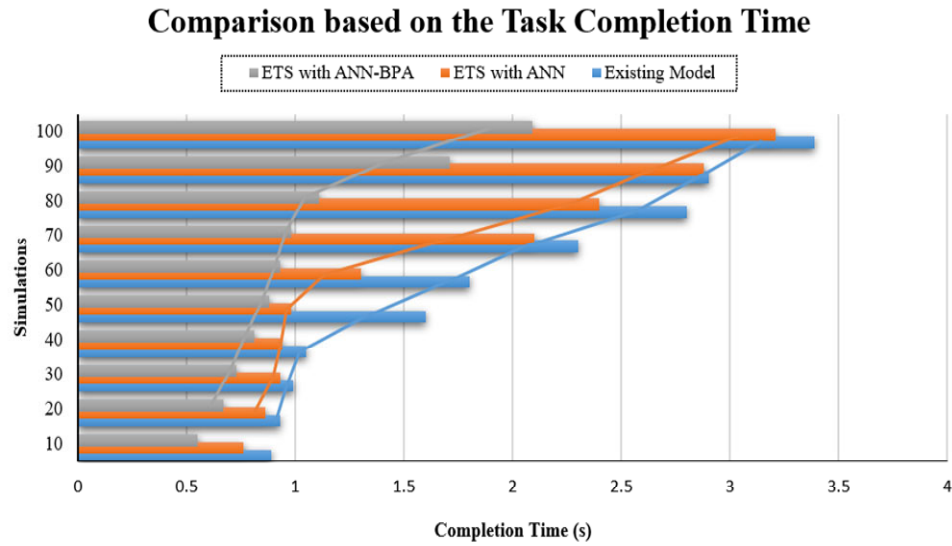
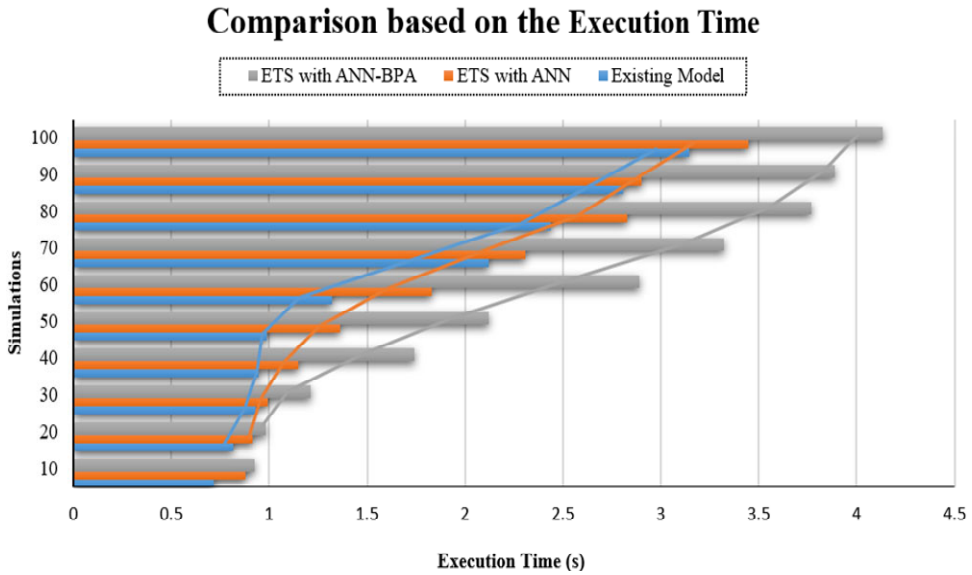


Table 6 Comparison based on the execution time

Simulations	Existing model (Zhu et al. 2021)	ETS with ANN	ETS with ANN-BPA
10	0.715	0.881	0.926
20	0.817	0.913	0.983
30	0.933	0.992	1.21
40	0.944	1.15	1.74
50	0.987	1.36	2.12
60	1.32	1.83	2.89
70	2.12	2.31	3.32
80	2.44	2.83	3.77
90	2.809	2.9	3.89
100	3.146	3.447	4.131

Figure 17 Comparison based on the execution time (see online version for colours)

In this section, there is need to compare the proposed ETS model with ANN-BPA approach to the most advanced scheduling algorithms that are currently being utilised and also compare with simple ANN-based model. From Figures 14 to 17 and Table 3 to 6, it has been observed that the overall performance of the proposed ETS model with ANN-BPA is better than proposed model with ANN and existing work (Zhu et al., 2021). Main reason of improvement in results analysis is the utilisation of MFO with ANN-BPA because their capacity to efficiently manage complicated and nonlinear relationships within data during the task allocation. The concept of backpropagation is a key training algorithm for ANNs that enables the network to learn from its mistakes by iteratively adjusting the weights of connections between neurons to minimise the difference between the predicted outputs and ground truth data during the allocation process. This process allows ANNs to continuously improve their performance and accuracy over time, making them highly adaptable to balance the load on each VMs and reduce the problem of VM migration with reduction in over and under loaded VMs count. Unlike normal ANNs, which may struggle with convergence and local minima problems, ANNs with backpropagation can efficiently navigate through high-dimensional spaces and optimise ETS model parameters to achieve better generalisation and robustness allocation with the help of return properties by MFO algorithm according to fitness. Additionally, backpropagation facilitates the training of deeper and more complex neural network architectures, enabling the extraction of more meaningful patterns of VMs according to the time from raw data and enhancing the network's ability to capture intricate patterns and relationships. Overall, ANN-BPA represents a powerful and versatile tool for solving a task scheduling problem in balanced manner.

5 Conclusions and future work

The foremost intention of this research is to develop ANN-BPA-based ETS model in cloud to perform task scheduling with a low-cost well-known MFO algorithm to reduce the amount of energy consumption as well as time spent waiting for cloud computing resources and balancing them. The suggested ETS model with ANN-BPA constitutes a robust and efficient work scheduling solution. By tackling issues such as load balancing, work distribution, and energy efficiency, it significantly enhances the performance of cloud-based task execution systems. The amalgamation of MFO with ANN-BPA enhances generalisation, flexibility, and resource optimisation, making it an appropriate methodology for practical applications. The suggested ETS model with ANN-BPA consistently demonstrated enhanced energy efficiency across all simulations. In ten simulations, energy consumption reduced from 349 units (existing model) to 252 units (ETS with ANN-BPA). In 100 simulations, energy consumption decreased from 658 units (existing model) to 535 units (ETS with ANN-BPA). The decrease underscores the capacity of ANN-BPA to improve resource use and equilibrate server loads efficiently. The ANN-BPA-based model for task scheduling is outperformed by existing approach as well as simple ANN-based model, which increases resource utilisation by 7.54% and decreases the time by 0.6 s. Therefore, our solution is superior for task scheduling in cloud using the concept of AI with optimisation mechanism. In the future, cloud computing will undergo a revolutionary change as a result of the development of autonomous task scheduling systems that are capable of self-learning and self-optimisation mechanisms. The dynamic workload changes that occur in real time will be adapted to by these systems, which will eliminate the need for human intervention and ensure optimal resource allocation and job execution.

References

- Ahmad, M.O. and Khan, R.Z. (2020) 'An efficient load balancing scheduling strategy for cloud computing based on hybrid approach', *International Journal of Cloud Computing*, Vol. 9, No. 4, pp.453–469.
- Ahmed, A., Adnan, M., Abdullah, S., Ahmad, I., Alturki, N. and Menzli, L.J. (2024) 'An efficient task scheduling for cloud computing platforms using energy management algorithm: a comparative analysis of workflow execution time', *IEEE Access*, Vol. 12, pp.34208–34221, DOI: 10.1109/ACCESS.2024.3371693.
- Amer, A.A., Talkhan, I.E., Ahmed, R. and Ismail, T. (2022) 'An optimized collaborative scheduling algorithm for prioritized tasks with shared resources in mobile-edge and cloud computing systems', *Mobile Networks and Applications*, Vol. 27, pp.1444–1460, <https://doi.org/10.1007/s11036-022-01974-y>.
- Arri, H.S. and Singh, R. (2021) 'Energy optimization-based optimal trade-off scheme for job scheduling in fog computing', in *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, March, pp.551–558.
- Bindra, S.S. and Aggarwal, A. (2024) 'Security in cyber physical systems: transformation and challenges', *Journal of Autonomous Intelligence*, Vol. 7, No. 4, pp.1–19.
- Kaur, S., Singh, J. and Bharti, V. (2024) A comparative study of optimization based task scheduling in cloud computing environments using machine learning', in *Proceedings – 2024 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, ICICV 2024*, pp.731–74.

- Khan, Z.A., Aziz, I.A., Osman, N.A.B. and Nabi, S. (2024) 'Parallel enhanced whale optimization algorithm for independent tasks scheduling on cloud computing', *IEEE Access*, Vol. 12, pp.23529–23548, DOI:10.1109/ACCESS.2024.3364700.
- Krishnasamy, K. G., Periasamy, S., Periasamy, K., Prasanna Moorthy, V., Thangavel, G., Lamba, R. and Muthusamy, S. (2023) 'A pair-task heuristic for scheduling tasks in heterogeneous multi-cloud environment', *Wireless Personal Communications*, Vol. 131, No. 2, pp.773–804.
- Li, F., Liao, T.W. and Zhang, L. (2019a) 'Two-level multi-task scheduling in a cloud manufacturing environment', *Robotics and Computer-Integrated Manufacturing*, September 2018, Vol. 56, pp.127–139, <https://doi.org/10.1016/j.rcim.2018.09.002>.
- Li, F., Zhang, L., Liao, T.W. and Liu, Y. (2019b) 'Multi-objective optimisation of multi-task scheduling in cloud manufacturing', *International Journal of Production Research*, Vol. 57, No. 12, pp.3847–3863, <https://doi.org/10.1080/00207543.2018.1538579>.
- Modh, R.M., Patel, M.B. and Patel, J.N. (2022) 'An extended scheduling of mobile cloud using MBFD and SVM', in *Smart Intelligent Computing and Applications, Proceedings of Fifth International Conference on Smart Computing and Informatics (SCI 2021)*, April, Vol. 1, pp.451–461, Springer Nature Singapore, Singapore.
- Mohammed, M.H. (2021) 'Bio-inspired approach and integrity check mechanism for secure data storage in multi-cloud environment', *Journal of Ambient Intelligence and Humanized Computing*, pp.1–9, <https://doi.org/10.1007/s12652-020-02757-7>.
- Mubeen, A., Ibrahim, M., Bibi, N., Baz, M., Hamam, H. and Cheikhrouhou, O. (2021) 'Alts: an adaptive load balanced task scheduling approach for cloud computing', *Processes*, Vol. 9, No. 9, pp.1–15, <https://doi.org/10.3390/pr9091514>.
- Nematollahi, M., Ghaffari, A. and Mirzaei, A. (2023) 'Task and resource allocation in the internet of things based on an improved version of the moth-flame optimization algorithm', *Cluster Computing*, Vol. 27, No. 2, pp.1775–1797.
- Pang, S., Li, W., He, H., Shan, Z. and Wang, X. (2019) 'An EDA-GA hybrid algorithm for multi-objective task scheduling in cloud computing', *IEEE Access*, Vol. 7, pp.146379–146389, <https://doi.org/10.1109/ACCESS.2019.2946216>.
- Pirozmand, P., Hosseinabadi, A.A.R., Farrokhzad, M., Sadeghilalimi, M., Mirkamali, S. and Slowik, A. (2021) 'Multi-objective hybrid genetic algorithm for task scheduling problem in cloud computing', *Neural Computing and Applications*, Vol. 33, No. 19, pp.13075–13088, <https://doi.org/10.1007/s00521-021-06002-w>.
- Shukri, S.E., Al-Sayyed, R., Hudaib, A. and Mirjalili, S. (2021) 'Enhanced multi-verse optimizer for task scheduling in cloud computing environments', *Expert Systems with Applications*, Vol. 168, p.114230, <https://doi.org/10.1016/j.eswa.2020.114230>.
- Singh, J., Bajaj, R. and Kumar, A. (2021) 'Scaling down power utilization with optimal virtual machine placement scheme for cloud data center resources: a performance evaluation', in *2021 2nd Global Conference for Advancement in Technology (GCAT)*, IEEE, October, pp.1–6.
- Sreenu, K. and Sreelatha, M. (2019) 'W-Scheduler: whale optimization for task scheduling in cloud computing', *Cluster Computing*, Vol. 22, No. 1, pp.1087–1098.
- Surjeet, K., Sabyasachi, P. and Ranjan, A. (2021) 'A particle swarm and ant colony optimization based load balancing and virtual machine scheduling algorithm for cloud computing environment', *Turkish Journal of Computer and Mathematics Education*, Vol. 12, No. 11, pp.3885–3898, <https://doi.org/10.17762/turcomat.v12i11.6504>.
- Taghinezhad-Niar, A. and Taheri, J. (2022) 'Reliability, rental-cost and energy-aware multi-workflow scheduling on multi-cloud systems', *IEEE Transactions on Cloud Computing*, Vol. 11, No. 3, pp.2681–2692.
- Zhu, Q.H., Tang, H., Huang, J.J. and Hou, Y. (2021) 'Task scheduling for multi-cloud computing subject to security and reliability constraints', *IEEE/CAA Journal of Automatica Sinica*, Vol. 8, No. 4, pp.848–865, <https://doi.org/10.1109/JAS.2021.1003934>.