

Original Article

Deriving a Multi-Objective Function for Load Balancing in Cloud Environment Via Adaptive Bayesian Belief Network with Ridge Regression-based Load Prediction

Hari Shankar Punna¹, Arif Mohammad Abdul²

^{1,2}Department of Computer Science and Engineering, GITAM (Deemed to be University), Hyderabad, Telangana, India.

¹Department of Computer Science and Engineering (Data Science), CVR College of Engineering, Hyderabad, Telangana, India.

¹Corresponding Author: hpunna@gitam.in

Received: 13 April 2026

Revised: 24 August 2026

Accepted: 29 August 2026

Published: 27 October 2026

Abstract - Cloud infrastructures require intelligent workload distribution mechanisms to maintain optimal throughput, control power consumption, and ensure efficient resource utilization. Traditional load balancing approaches fail to address several persistent issues - including workload unpredictability, delayed response cycles, server overutilization, and suboptimal task scheduling. This study proposes an Adaptive Bayesian Belief Network-Ridge Regression framework, termed MRA-SAO-ABBNRR, which employs a Modified Random Attribute Snow Ablation Optimizer. The proposed model integrates Bayesian Belief Networks with Ridge Regression to forecast future workload patterns from historical cloud telemetry. Overfitting is mitigated through correlation analysis across input attributes. Additionally, a modified random attribute snow ablation optimization scheme is used to fine-tune workload prediction and to support a multi-objective task allocation strategy spanning resource utilization, response time, makespan, scalability, and energy efficiency. All experiments were conducted in MATLAB 2020a, and benchmark comparisons were drawn against four existing models, namely Convolutional Neural Network-Long Short-Term Memory, Variable Model Matching Intelligent Self-Distillation, Ensemble Prediction and Risk Uncertainty Assessment, and Bayesian Belief Networks-Ridge Regression. The proposed framework delivered superior prediction accuracy alongside more effective load balancing. It recorded the lowest SMAPE of 0.2 when paired with the sigmoid activation function, outperforming every baseline. The model also demonstrated reduced utilization of compute resources, lower operational expenditure, and decreased energy demand.

Keywords - Bayesian belief network, Cloud load balancing, Multi-objective optimization, Ridge regression, Snow ablation optimizer.

1. Introduction

In the environment of cloud computing, the transmission is done through the internet. Informatics, preservation, records, and networking are a few examples of these resources [9]. The real-world implementation of a cloud-based system has presented several challenges. These include scheduling, safety, privacy, and resource finding [10]. In addition to providing facilities to reserve material internet rather than in computer files, the cloud service enables users to access data and apps from system files via an internet connection. Additionally, depending on the needs of the user, the cloud services offer enormous storage capacities for storing documents. Many software companies rely on computing for utility purposes, which is made possible by cloud computing [11, 12]. Cloud service providers are authorized to give these resources to cloud customers, and they are often maintained in virtual form. With the available virtual resources, the service

provider offers the services to the user. In cloud computing environments, VM migration is an essential procedure that allows server consolidation, workload weighing, and maintenance activities without interruptions in service [13]. Considering its benefits, VM migration is a difficult Endeavour with several obstacles, including load distribution, the energy economy, and security levels. Thus, it proved necessary to balance the cloud's burden.

Addressing these distribution hurdles requires routing incoming computing tasks efficiently across multiple execution nodes [14]. Implementing robust workload scheduling directly elevates overall system efficiency and throughput [15]. Effective allocation policies reconcile provider cost constraints with user satisfaction standards. Within cloud architectures, scheduling mechanisms operate across physical infrastructure to govern task routing between



compute nodes [16]. While static assignment policies perform adequately in uniform hardware setups, adaptive scheduling models are required for mixed infrastructure topologies where workloads vary dynamically [17, 18]. Consequently, modern frameworks deploy sophisticated balancing strategies to ensure uniform task distribution across data center resources [19]. Consequently, modern frameworks deploy sophisticated balancing strategies to prevent server bottlenecks across the data center. Both cloud service providers and users should prioritize load balancing so that the earlier group seeks to maximize resource utilization and throughput while minimizing expenses, delays, application completion times, and VM makespan [20].

Minimization of energy consumption, balancing the load, and managing the migration of VM cause challenges in the cloud [21]. To improve the cloud services, trust-based clustering and agents are proposed in the prior works. However, [22], the high overload issues were not handled by this model, and they do not even provide the finest optimal results in the load balancing operation [23]. The high data overload within the network does not provide the finest outstanding results in load balancing [24]. If the same process is continued, the performance of the VM is greatly affected [25]. Secure data sharing during migration, reducing energy consumption, and workload balancing are not attained by the traditional methods. Heuristic techniques such as Particle Swarm Optimization, Butterfly Optimization, and Honey Bee Optimization deliver strong performance for resource scheduling in distributed computing. However, slow convergence speeds across these metaheuristic models frequently constrain overall operational efficiency. To minimize task processing delays and overcome these computational bottlenecks, an advanced prediction and resource distribution framework is established.

The highlights of the research work are presented in the points below:

- The proposed architecture incorporates advanced workload forecasting alongside multi-server resource scheduling to maintain stable operation across cloud data centers. By continuously calculating pending resource demands and routing incoming network traffic across distributed nodes, the model maximizes system performance. Furthermore, this systematic allocation strategy accommodates large-scale, heterogeneous consumer requests while upholding high standards of service reliability.
- The MRA-SAO optimization algorithm tunes internal hyperparameters, including learning rates, hidden layer counts, and training epochs, to reduce prediction error metrics such as Root Mean Square Error and Mean Absolute Error. Conversely, the algorithm governs task allocation across available servers to improve system

scalability and reduce total makespan during resource scheduling.

- The ABBN-RR forecasting engine integrates Bayesian probability networks with regularized regression to anticipate cloud resource demands. Within this architecture, the probabilistic component manages uncertainty inherent in historical telemetry, while the regression mechanism suppresses multicollinearity among correlated attributes. This combined formulation prevents parameter overfitting, sharpens forecasting precision, and delivers high predictive performance while requiring minimal computational overhead.
- Optimization functions including response time, makespan, resource utilization, energy efficiency, and system scalability, are governed through the MRA-SAO algorithm to manage volatile cloud workloads effectively. Satisfying multiple constraints simultaneously resolves intricate load-balancing bottlenecks within distributed architectures. Spreading computation evenly across available infrastructure nodes prevents server saturation and safeguards the network against operational failures.
- The rest of the research work is arranged in the upcoming parts: In Part 2, the literature survey is discussed in detail. The architecture of load balancing in a cloud system with its demand is presented in Part 3. The description of utilized optimization and deep learning algorithms used for performing load prediction is presented in Part 4. Part 5 introduces the multi-objective scheduling framework designed for distributed cloud environments. The experimental evaluation is discussed in Part 6. The conclusion is presented in Part 7.
- A lot of achievements have already been made in cloud load balancing research area, yet there are still some unsolved issues. Current approaches focus mainly on either load prediction or load balancing, and they cannot take any preventive measures concerning task scheduling based on workloads. Moreover, typical optimization algorithms contain just a certain number of factors, for example, only execution time and power consumption, without considering any others, including scalability, resource allocation, and task completion time. While using machine learning techniques, especially deep and reinforcement learning ones, can provide higher flexibility, it will lead to a very high computation cost and a huge amount of data for training. That is why there is needed an approach which will be capable of predicting workloads and provide multi-objective load balancing. To solve the problem, a combination of Adaptive Bayesian Belief Network and Ridge Regression Model, along with Multi-Criteria Resource Allocation and Self-Adaptive Optimization Algorithm, may be applied. The primary aims of this architecture focus on elevating workload forecasting accuracy while jointly optimizing resource utilization, task completion velocity, energy efficiency, and overall scalability.

2. Background Review and Contextual Studies

2.1. Overview of Prior Research

Recent research in cloud computing has extensively focused on load balancing and workload prediction to improve resource utilization, reduce energy consumption, and enhance service quality. Existing approaches mainly employ reinforcement learning, swarm intelligence, deep learning, and hybrid optimization techniques.

Although these methods report improvements in individual performance metrics, a critical examination of the literature reveals that most frameworks remain limited by high computational overhead, inadequate scalability, poor adaptability to dynamic workloads, or insufficient integration between workload prediction and scheduling mechanisms.

Pradhan et al. [1] formulated a combined heuristic approach utilizing deep reinforcement learning and parallel swarm techniques for cloud scheduling. Although this reward-driven strategy successfully shortened makespan and lowered power consumption, relying heavily on iterative training loops elevated memory overhead and computational complexity.

Furthermore, policy convergence proved unstable during volatile workload shifts, restricting real-time deployment viability. The investigation also omitted an integrated predictive mechanism, reducing its capacity for proactive workload management.

To improve optimization efficiency, Reshan et al. [2] developed a combined metaheuristic model pairing Grey Wolf Optimization with Particle Swarm Optimization (GWO-PSO) for task distribution. While the integration of swarm intelligence techniques improved convergence speed and reduced response time, the framework primarily performs reactive scheduling based on current system conditions.

The absence of workload forecasting restricts its ability to anticipate workload variations, which is critical in highly dynamic cloud infrastructures. Furthermore, the study focuses mainly on optimization performance without addressing long-term scalability and adaptive resource management.

Brahmam and Anand [3] developed the VMMISD framework for energy-efficient virtual machine migration using clustering-based optimization and security-aware mechanisms. Although the model achieved improved deadline-hit ratios and lower storage costs, its practical implementation involves substantial complexity due to VM clustering, migration management, and security integration.

Moreover, the framework exhibits limited scalability when deployed in large-scale distributed cloud environments. The study primarily emphasizes migration efficiency rather than comprehensive multi-objective scheduling optimization.

Kaviarasan et al. [4] formulated an enhanced feline-inspired heuristic known as the Improved Lion Optimization Algorithm (ILOA_LB) to manage cloud task distribution. The algorithm demonstrated better throughput and fault tolerance as the number of cloud nodes increased. Nevertheless, the framework incurred high energy consumption and lacked a predictive workload analysis component. As a result, the scheduling mechanism reacts only after workload changes occur, reducing efficiency during sudden workload spikes and increasing resource imbalance in dynamic cloud scenarios.

Similarly, Udayasankaran and Thangaraj [5] introduced the EPRUA framework to minimize energy utilization through efficient traffic distribution across virtual machines. Although the method reduced processing time and enhanced VM performance, the framework requires complicated resource configuration and extensive setup procedures. Such operational complexity limits its adaptability in heterogeneous and large-scale cloud environments. Furthermore, the investigation omits joint multi-metric tuning for essential quality-of-service attributes like power conservation, latency reduction, and network scalability.

In a study conducted by Gerald et al. [6], a Fruitfly Transfer Learning Architecture (FTLA) was formulated to manage job scheduling and virtual machine resource sharing. The framework improved execution efficiency and reduced overloading risks; however, the scheduling process still experiences high execution time under increasing workloads. More importantly, the model lacks an integrated workload prediction strategy, preventing proactive decision-making for resource allocation and limiting adaptability during dynamic workload fluctuations.

Deep learning-based approaches have also gained considerable attention for workload prediction and resource provisioning. Simaiya et al. [7] formulated a combined convolutional neural network and long short-term memory architecture coupled with Particle Swarm Optimization and Genetic Algorithm mechanisms (PSO-GA) to resolve resource under-provisioning and over-provisioning dilemmas. The framework improved provisioning accuracy and reduced idle power consumption. Despite these advantages, the model requires substantial training time and computational resources, making real-time deployment difficult in large cloud environments. Furthermore, the framework primarily focuses on prediction accuracy without comprehensively optimizing scheduling objectives such as makespan, scalability, and response time.

Similarly, Kaur et al. [8] established a neural network-driven architecture known as the Deep Learning-based Deadline-constrained, Provisioning and Load Balancing (DLD-PLB) model to manage workflow scheduling across distributed cloud nodes. Although the approach reduced response time and operational cost, the training complexity of

the deep learning architecture limits its applicability in rapidly changing cloud environments. The study also lacks an adaptive optimization mechanism capable of balancing multiple scheduling objectives simultaneously under uncertain workload conditions.

From the above studies, it can be critically observed that existing cloud load balancing frameworks suffer from three major research limitations. First, workload prediction and task scheduling are generally treated as independent processes, resulting in poor coordination between forecasting and resource allocation decisions. Second, most existing methods optimize only one or two objectives, whereas practical cloud systems require simultaneous tuning of latency minimization, completion velocity, power conservation, network expandability, and infrastructure efficiency. Third, several optimization and deep learning models exhibit high computational complexity, slow convergence, and limited adaptability under dynamic workloads, thereby restricting practical implementation across distributed enterprise data centers.

Consequently, developing a unified and computationally efficient architecture that merges proactive workload forecasting with adaptive multi-objective optimization remains essential. The proposed MRA-SAO-ABBNRR framework addresses these limitations by integrating Adaptive Bayesian Ridge Regression-based workload prediction with an enhanced metaheuristic scheduling mechanism. Unlike existing reactive approaches, the proposed framework enables proactive load balancing while simultaneously optimizing resource utilization, response time, energy consumption, scalability, and makespan, thereby improving overall cloud performance and adaptability.

2.2. Critical Research Gap

A systematic analysis of the current literature shows that there are three recurring gaps in the field. The first gap is that the two challenges of predicting workload and scheduling tasks have always been considered independently, thereby hindering a framework's ability to use its predictions for decision-making regarding task scheduling. The second gap is that most of the current optimization approaches have only considered optimizing one or two metrics, while practical applications require the joint optimization of resource utilization, makespan, response time, energy consumption, and scalability.

2.3. Differentiation of the Proposed Method

Unlike other models, such as DRLPPSO [1] and CNN-LSTM [7], which entail significant computation and consider prediction and scheduling as independent processes, the new framework utilizes the probabilistic workload prediction technique based on ABBN-RR in the scheduling process through the MRA-SAO framework. As opposed to previous models such as VMMISD [3] and ILOA_LB [4], which focus on optimizing specific goals, the proposed architecture simultaneously suppresses latency, power consumption, and makespan while maximizing infrastructure throughput and expandability. In addition, the new MRA-SAO model uses a contemporary random attribute strategy that improves search diversity and convergence rate compared to traditional Snow Ablation Optimizer variations. This model exhibits superior performance compared to BBN-RR, CNN-LSTM, EPRIJA, and VMMISD algorithms, with an SMAPE value of 0.2 as opposed to 1.6, 1.0, 0.9, and 1.2, respectively. An overview detailing the operational characteristics and limitations of prior cloud task allocation frameworks is summarized in Table 1.

Table 1. Operational characteristics and constraints of traditional load balancing architectures

Reference [Citation]	Allocation Algorithm	Key Advantages	Operational Limitations
Pradhan et al. [1]	DRLPPSO	<ul style="list-style-type: none"> This model effectively reduces energy consumption and makespan time. It handles a large number of tasks with the help of a parallel processing approach. 	<ul style="list-style-type: none"> It requires high computational power and memory. Resource allocation ability is poor.
Reshan et al. [2]	GWO-PSO	<ul style="list-style-type: none"> Enhances convergence velocity and minimizes total latency. 	<ul style="list-style-type: none"> Demonstrates reduced predictive precision.
Brahmam and Anand [3]	VMMISD	<ul style="list-style-type: none"> It effectively identifies new security threats and attains better deadline-hit ratio outcomes. The storage costs are very low. 	<ul style="list-style-type: none"> Implementation complexity is high. Scalability is less.
Kaviarasan et al. [4]	ILOA_LB	<ul style="list-style-type: none"> It has high fault tolerance and better throughput in load balancing. It efficiently handles high traffic even when the node count increases. 	<ul style="list-style-type: none"> Energy consumption is high.
Udayasankaran and Thangaraj [5]	EPRUA	<ul style="list-style-type: none"> Cloud center's energy utilization is minimized by this model. 	<ul style="list-style-type: none"> Managing dynamic resource allocation is complicated.

		<ul style="list-style-type: none"> It enhances the performance of virtual machines and decreases the processing time. 	<ul style="list-style-type: none"> It needs more time to configure cloud servers and virtual machines.
Gerald et al. [6]	FbTL	<ul style="list-style-type: none"> It reduces the risk of overloading. It allocates enough resources for the task execution. 	<ul style="list-style-type: none"> Execution time is high.
Simaiya et al. [7]	CNN-LSTM	<ul style="list-style-type: none"> It solves the under-provisioning and over-provisioning issues. It reduces the idle power usage of data centers. 	<ul style="list-style-type: none"> The computational resources requirement for the training is high. It does not give importance to security and privacy.
Kaur et al. [8]	HDD-PLB	<ul style="list-style-type: none"> It efficiently reduces the response time and cost. 	<ul style="list-style-type: none"> Model training time is high.

3. Structural Design and Requirements of Cloud Load Balancing

3.1. Operational Demand for Resource Balancing in Distributed Networks

Enterprise computing paradigms increasingly rely on distributed infrastructure to address complex digital processing requirements. System imbalances routinely emerge when localized network congestion develops due to expanding global user demand. As consumer requests traverse the network, cloud providers deliver diverse computational services to maintain operational continuity. The commercial sector uses parallel computing, distributed computing, and grid computing since various services are offered by cloud computing services. However, the biggest issues brought by the growth of cloud computing systems are security and scheduling. The growing volume of daily requests in cloud computing can cause load-balancing problems and potentially strain the cloud data center. Load balancing serves as a primary mechanism to mitigate these issues within cloud architectures. Physical servers consume electrical energy when executing specific tasks in the cloud environment.

Here, more energy is consumed if the server has several workloads. So, the demand for load balancing is increased for handling the load among the VM. Proposed Load Balancing Framework in Cloud Architecture. The communication services are enriched by the cloud since it offers a smart platform for digital applications. Over the internet, the user can receive virtualized, dynamic, and scalable computing resources leveraging cloud technology. Distributing workloads across cloud infrastructures constitutes an NP-hard problem. The prior models developed for the scheduling operation take more uptime, and they are subjected to more power consumption. Further, the conventional models for the load-balancing process cause more response time because of the ineffective load-balancing capability, and they do not ideally distribute the resources within the cloud system. By observing these issues, the proposed framework formulates a new model that predicts the workload and schedules it to the physical machines of the cloud system for fast processing of tasks. The overall illustration of the suggested load prediction with scheduling models is displayed in Figure 1.

In order to handle the load among the VM in the cloud, the proposed framework develops an intelligent model for workload estimation and load balancing. This helps to complete tasks faster and use less energy in the server facilities. Further workload distribution between virtual machines and hardware nodes is balanced by the proposed load balancing strategy since the nodes have equal load, because the task is allocated to the cloud data center with the help of the proposed model. The designed workload prediction and better resource utilization are provided by load balancing also it reduces the carbon footprint of the data center also the cooling requirements of the machine are reduced by the suggested strategy.

The energy consumption and execution duration are minimized by the proposed framework while simultaneously enhancing cloud-hosted virtual instance performance. Initially, the workload prediction is executed to enhance the efficiency and the operational cost of the cloud. Better results in the load balancing are provided by the efficient load prediction model. Here, the ABBN-RR is realized to predict the workload in the cloud environment, deployed to optimize infrastructure resource utilization. The proposed ABBN-RR represents a key milestone for workload forecasting within cloud center architectures.

Within this ABBN-RR mechanism, BBN and RR collaborate to yield precise forecasting outcomes, generating workload distribution estimates where the discrepancy between predicted and empirical values is quantified. The predicted workload accuracy is enhanced upon the incorporation of the BBN and RR with the proposed ABBN-RR.

The proposed ABBN-RR uses the MRA-SAO to further enhance the accuracy of the workload prediction as it tunes hidden neurons, learning rate, and steps per epoch, which reduces the MAE and RMSE values in the workload prediction. The predicted workload is used for making informed decisions regarding the balancing of load in the cloud system. Here, the MRA-SAO aims to distribute tasks across the cloud architecture. Individual user job requests are submitted to the entity. The submitted jobs are scheduled and

allocated by the entity. The proposed MRA-SAO-based load balancing is executed during the run time when server utilization exceeds predefined thresholds. The proposed MRA-SAO redistributes demands from overtaxed nodes based on resource utilization, makespan, response time, energy, and scalability factors. The available capacity of the server and its workload are reverted back to the broker. Here,

the waiting task on the server is reallocated by the server for their execution. The upper bound on the anticipated load of the server is considered by the MRA-SAO for preventing the overloading of the server within the cloud system. Finally, the recommended framework has been compared with the earlier models to prove its effectiveness in load prediction and balancing operations.

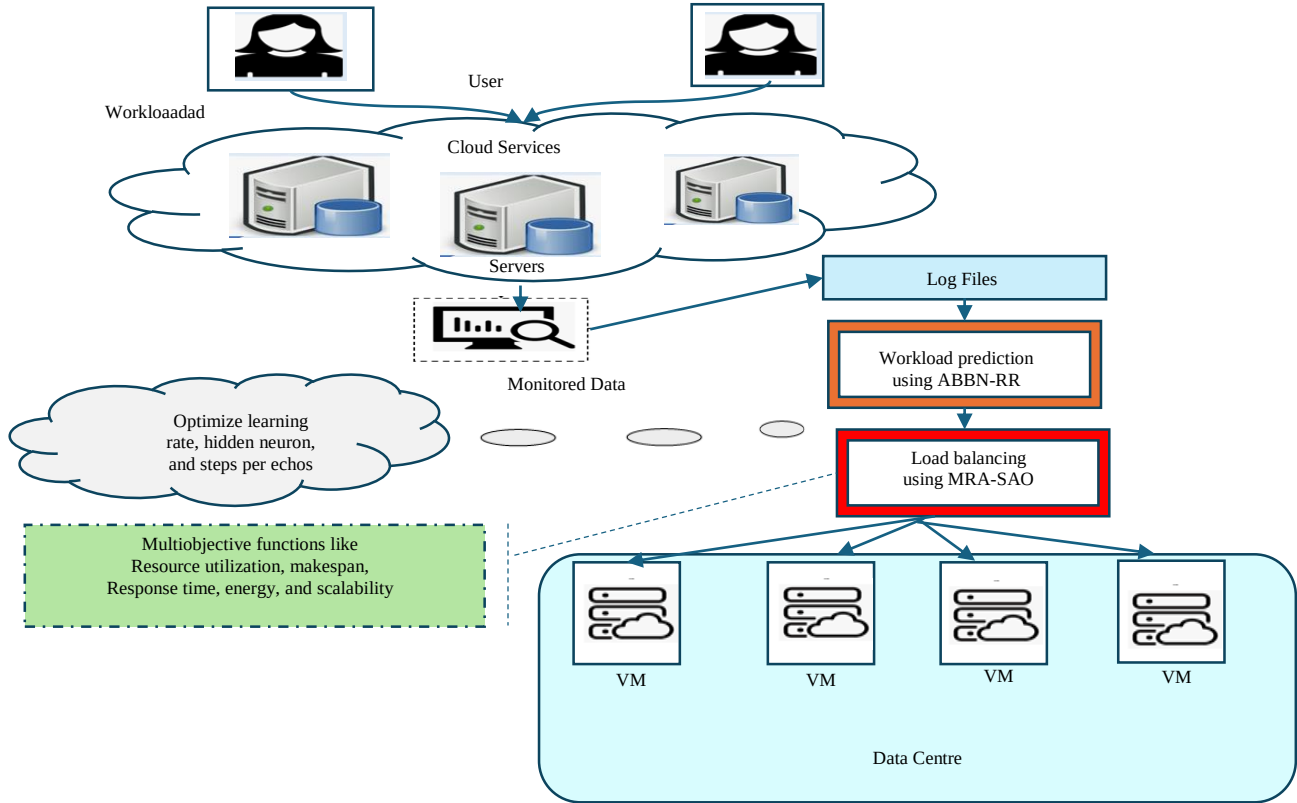


Fig. 1 Overall illustration of the suggested load prediction with balancing models

4. Description of Utilized Optimization and Deep Learning Algorithms used for Performing Load Prediction

4.1. Designed MRA-SAO

The MRA-SAO is the independent load-balancing algorithm proposed for enhancing the processing capability of the cloud system by preventing the underloading or overloading of the server. The MRA-SAO-based load balancing minimizes execution duration and accelerates overall cloud infrastructure performance. The MRA-SAO is designed from the conventional SOA [26]. The SOA is more suitable for the global optimization process however; the local confinement is the greatest issue in the SOA. Some of the powerful confinements of the SOA are exploitation and exploration. However, maintaining the equilibrium between the exploitation and exploration dynamics is constrained by the SOA. Thus, the idea of improving the random number is suggested to avoid these difficulties in the conventional SOA. The current, worst, and mean fitness are adopted for calculating the new random number. The MRA-SAO tunes in

the hidden neuron, learning rate, and epoch to enhance the accuracy while minimizing the MAE and RMSE in the load prediction operation. Also, the MRA-SAO is adopted in the load balancing operation to provide a feasible solution based on objective functions, including resource utilization, makespan, response time, energy, and scalability. The proposed MRA-SAO-based load distribution aims to distribute the load among the user, which is the main element of load balancing.

When compared to the usual SAO, the proposed MRA-SAO found better results in the balancing exploration and exploitation phases. Throughout the search phase, the proposed MRA-SAO thrives on attaining the global optimal results. The MRA-SAO-based load balancing is easy to implement, and it is a less expensive process. The reliability of each node in the cloud system is improved by performing load balancing using the MRA-SAO, so it provides enhanced reliability and better throughput. Minor tasks experience negligible queuing delays under the introduced MRA-SAO

allocation mechanism. The formula for calculating the new random number is given in Equation (1).

$$\kappa = \frac{fr+wr}{fr+wr+lr} \quad (1)$$

Here, the random number is expressed as κ and it is upgraded in the Equation (2) of the SAO. The current and worst fitness factors are expressed as fr , wr . The mean fitness is denoted as lr .

$$A_j(u+1) = E(u) + CN_j(u) \otimes \left(\frac{\kappa \times}{H(u) - A_j(u)} \right) + (1 - \kappa) \times \left(\frac{\tilde{A}(u)}{-A(u)} \right) \quad (2)$$

The present best solution is $H(u)$ that the vector is specified as $CN_j(u)$, during the iteration u , the j^{th} individual is represented as $A_j(u)$, the random individual selected from the elite is represented as $E(u)$, and the whole swarm's centroid position is denoted as $\tilde{A}(u)$. The proposed MRA-SAO iteratively explores the search space, and it can effectively solve the problems. The near-optimal solution is determined by the MRA-SAO as it is dynamically adapted to the change in environmental conditions. Algorithm 1 outlines the procedural steps for the proposed MRA-SAO method.

Algorithm 1: Developed MRA-SAO	
Input: Hidden Neuron F_l , Learning Rate Q_w , and Steps Per Epochs A_q , Initialized Attributes	
Output: Optimal Hidden Neuron $F_{l*}^{mra-soa}$, Refined Learning Rate $Q_{w*}^{mra-soa}$, and Optimized Steps Per Epochs $A_{q*}^{mra-soa}$, tasks To be assigned to the sever $d_{q*}^{mra-soa}$	
Evaluate candidate fitness scores for all search agents.	
Identify the current global best solution.	
While the terminal condition not fulfilled	
	The value of the snowmelt rate is determined.
	Determine the new random number using Equation (1).
	Split the search population into two distinct subsets.
	For every individual search agent:
	Revise agent positions using Equation (2).
	End for
	Re-evaluate individual fitness values.
	Refresh the optimal solution record.
End while	
Output the finalized optimal solution	
Stop	

4.2. The Bayesian Belief Network

The BBN [27] is a computationally efficient model for predicting the load in the cloud system. This model combines the different knowledge sources for predicting the maximum load within the cloud system. The Reverend Thomas Bayes theorem is associated with the artificial intelligence method for developing the BBN. The directed acrylic graph is present

in the BBN. Here, the dependence between the linked variables is represented as arcs, and the variables are represented in the form of nodes. The conditional probabilities are used for expressing the strength of the influences. Equation (3) expresses the Bayes rules.

$$Q(c/b) = \frac{Q(b/c) \times Q(c)}{Q(b)} \quad (3)$$

The term b 's probability is expressed as $Q(b)$, the probability of b is given that occurred is represented as $Q(b/c)$, the hypothesis, evidence and experience are one of the complex ways for expressing Bayes' rules, and it is expressed in Equation (4).

$$Q(I/F, d) = \frac{Q(I/d) \times Q(F/I, d)}{Q(F/d)} \quad (4)$$

The probability of the hypothesis is $Q(I/F, d)$ and d , the prior probability of I is denoted as $Q(I/d)$, the likelihood is denoted as $Q(F/I, d)$, and the normalizing scaling factor is denoted as $Q(F/d)$.

4.3. Ridge Regression Network

The multicollinearity problems are effectively alleviated by the useful algorithm known as Ridge Regression (RR) [28]. The linear expression available within the RR model is indicated in Equation (5).

$$Z = Y\phi \quad (5)$$

The above expression is used for minimizing the residual, and it is expressed in Equation (6).

$$\|Z - Y\phi\|^2 \quad (6)$$

Here, the $o \times 1$ vector is expressed as Y , the $o \times q$ matrix is denoted as Y , the $q \times 1$ vector is represented as ϕ , and the Euclidean norm is denoted as. The non-unique solution is attained by the matrix Y since it is one type of singular yielding approach. In the above expression, the regularization term is added to attain the solution with desirable properties and it is expressed in Equation (7).

$$\|Z - Y\phi\|^2 + l^2 \|\phi\|^2 \quad (7)$$

The conditioning of the problem is improved by the above regularization and the explicit solution $\hat{\phi}$ is given in Equation (8).

$$\hat{\phi} = \hat{\phi}(l) = (Y'Y + lI)^{-1}Y'Z \quad (8)$$

Here, the positive number is expressed as l , which lies in between (0,1).

4.4. Load Prediction in Cloud Environment using Proposed ABBNRR

The ABBNRR is developed by integrating the BBN and RR models. The Bayes theorem is used by the BBN for getting accurate results in the load prediction process. The probability estimates are updated by the Bayes theorem. The relationship between the variables is determined by learning the relationship between the data using the BBN. The belief about the future loads is determined by this model. The BBN model analyzes the activity of the network to predict the load within the system. Further, the chance of overfitting is prevented by regularization in the RR. Further, the relationship between the predictors is modelled by the RR. The workload of the cloud system is measured by considering the historical data in the cloud system. The historical data is composed of memory usage, workload patterns, and other relevant indicators of the cloud. The diverse workload metrics and conditional dependencies of the metric are represented by the nodes of the BBN. Various factors and relationships among the workload factors in the cloud are captured by the BBN. Based on the historical data of the cloud system, the required factors for the BBN are fixed. The multicollinearity over the workload factor is addressed by the RR. The predictions are refined by the components of the RR. Initially, the BBN model learns the conditional probabilities to generate the feature regarding the workload of the cloud environment. Further, the features learned from the models are passed to the RR, which continuously learns the features and predicts the workload in the dynamic cloud system. The proposed MRA-SAO tunes the hidden neuron, learning rate, and epoch to maximize the

accuracy while lowering the RMSE and MAE. The objective function of the suggested ABBNRR-based load prediction model for the cloud environment is given in Equation (9).

$$QWS = \arg \min_{\{F_{l*}^{mra-soa}, Q_{w*}^{mra-soa}, A_{q*}^{mra-soa}\}} \left(\frac{1}{A} + RMSE + MAE \right) \quad (9)$$

Here, optimized hidden neuron $F_{l*}^{mra-soa}$ lies in between [5,255], optimized learning rate $Q_{w*}^{mra-soa}$ stuck in the middle of [0.01,0.99], and optimized steps per epochs $A_{q*}^{mra-soa}$ ranges between [100,500]. The accuracy, RMSE, and MAE are calculated using Equation (10).

$$A = \frac{er + et}{er + et + ui + uo} \quad (10)$$

Here, the “true positive and negative are expressed as er, et and false positive and negative” is ui, uo .

$$RMSE = \sqrt{\sum_{j=1}^o \frac{(\hat{q}_j - q_j)^2}{o}} \quad (11)$$

$$MAE = \sum_{j=1}^o \frac{(\hat{q}_j - q_j)}{o} \quad (12)$$

Here, the “actual value” is denoted as $\hat{q}_j$ and the “predicted value” is expressed as q_j . The structural representation of the proposed ABBNRR-based load prediction is given in Figure 2.

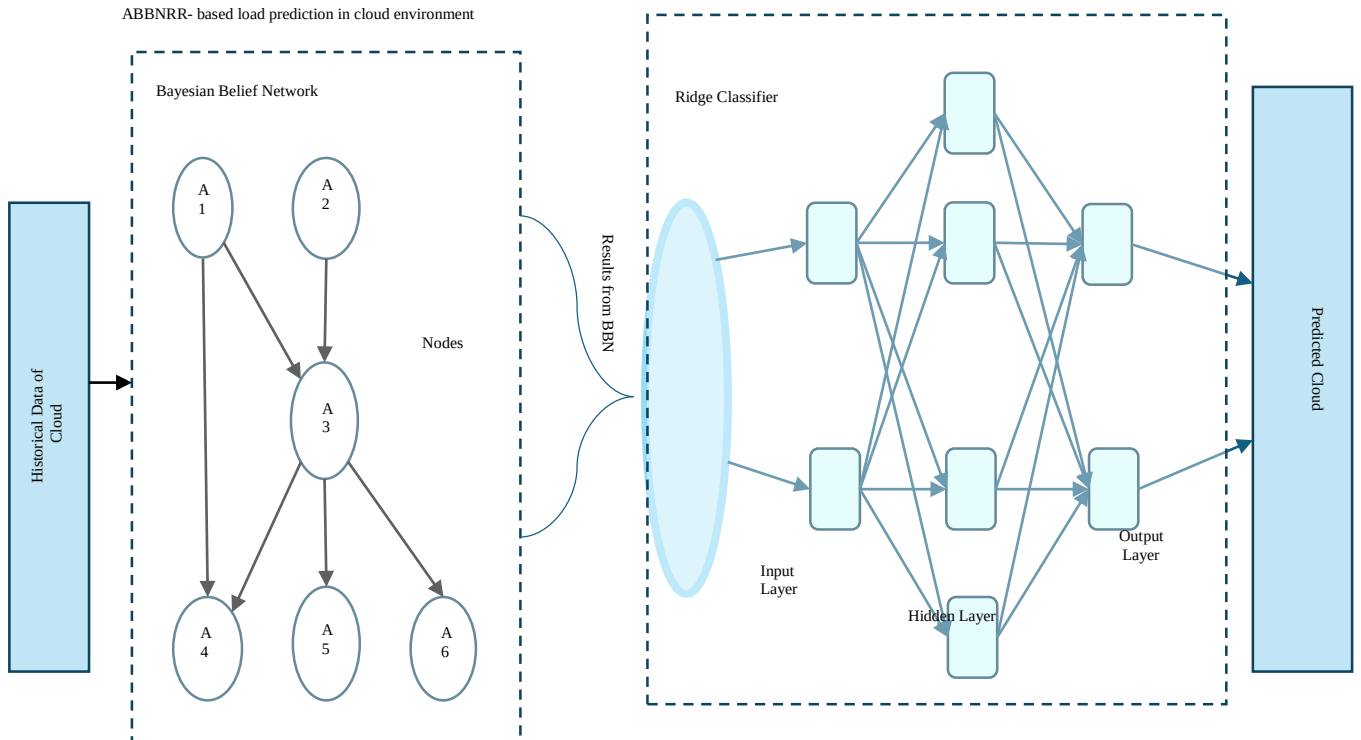


Fig. 2 Structural representation of the proposed ABBNRR-based load prediction

5. Multi-Objective-based Load Balancing Framework in Cloud Environment

5.1. Network Initialization

The network parameters like “Workload Attributes, Request Timestamp, Source IP Address, Destination IP Address/Service, Protocol Type, Request Size, Response Size, Request Type, Session or Connection ID, Inter-Arrival Time, Server ID, Server Capacity, Server Load, Server Status, Bandwidth Availability, Latency, Packet Loss Rate, Jitter, Network Topology, Concurrent Connections, Session Duration, Request Frequency, Workload, Task Attributes, Available memory, Available CPU, Processing time, Reaction time, CPU Clock Speed, Priority of all tasks, Memory requirement of all tasks, CPU requirement of all tasks, Completion time of all tasks” are initialized for the load balancing operation.

5.2. Load Balancing in Cloud Environment using Proposed MRA-SAO

Server capacity and task duration dictate load-balancing requirements. Task allocation accounts for node capacity, incoming job size, and pre-existing server load calculated via the proposed ABBNRR framework. Efficient resource distribution across cloud architectures relies on load balancing to maintain peak system performance. The proposed MRA-SAO optimizes cloud performance while efficiently distributing workloads across virtual machines. Conventional load balancing algorithms struggle with dynamic and complex workloads, making the advanced MRA-SAO framework essential for optimal outcomes in cloud operations. MRA-SAO effectively manages large user volumes and fluctuating workloads by adapting dynamically to demand variations, ensuring optimal resource allocation and maximum efficiency.

By distributing network traffic across multiple suitable servers, the framework minimizes response time and maximizes overall system performance. To prevent server congestion, MRA-SAO evaluates idle capacity and monitors the resource usage and health of each cloud server. Based on current load and server performance, the framework selects the optimal server. Users submit tasks to the broker, which in turn assigns them to the appropriate servers. If any server exceeds predefined threshold limits, MRA-SAO rebalances the workload on the congested node. Equation (13) defines the overall objective function of this load-balancing operation.

$$kjh = \arg \min_{\{d_{q^*}^{mra-soa}\}} \left(\frac{1}{RY} + \frac{1}{MK} + \frac{1}{SC} + RT + EC \right) + CAPY + LT \quad (13)$$

Here, the objective function is termed $askjh$, the tasks to be assigned to sever $d_{q^*}^{mra-soa}$, Resource Utilization RY , Makespan MK , Scalability SC , Response Time RT , and Energy Consumption EC , the capacity of the nodes $CAPY$, and length

of the task for the node is expressed as LT , they are explained in the upcoming sections. Figure 3 illustrates the architecture of the MRA-SAO-based load balancing framework.

The proposed methodology identifies overworked and under-utilized cloud nodes, redistributing tasks to achieve balanced workloads [4]. Since cloud users depend entirely on the platform for their computational needs, service providers must ensure equitable task management, thereby guaranteeing business continuity for end users.

Initially, the total count of cloud nodes is initialized using Equation (14), after which individual node fitness scores are evaluated via Equation (15) to measure processing efficiency.

$$O_{CLOUD} = [o_1, o_2, o_3, \dots, o_o] \quad (14)$$

$$g(O_{CLOUD}) = g(o_1, o_2, o_3, \dots, o_o) \quad (15)$$

Computing node fitness requires evaluating the volume of processed tasks alongside the total number of network nodes. The term y_{jk} denotes tasks allocated to each respective node, as formulated in Equation (16).

$$y_{jk} = \begin{cases} 1 & \text{if TASK } j \text{ is Allocated} \\ 0 & \text{else} \end{cases} \quad (16)$$

The capacity of the VM and the length of the task have been analyzed before the task is allocated to the VM.

$$CAPY = QF_n * QF_{mips} \quad (17)$$

The capacity of the nodes is represented as $CAPY$, each node's processing element is expressed as QF_n . The million instructions per second is represented as QF_{mips} for processing the element. The underutilized nodes are initially identified during the procedure of handling the task. In the determined underutilized nodes, most of the task is moved because of the course of action. The expression for identifying the underutilized nodes is expressed in Equation (18).

$$O'_{UN} = O_{UN} + \kappa(0,1) * QJ * (O_{UN} - I) \quad (18)$$

Thus, the closest neighbor nodes are represented as I , the performance improvement is represented as QJ , the old underutilized nodes are expressed as O_{UN} , and the new underutilized node is O'_{UN} .

The main steps of the load-balancing operation using the MRA-SAO are given as follows:

- Step 1: Initialize the number of nodes
- Step 2: Divide the cloud system as territory
- Step 3: Determine the fitness value of each node
- Step 4: The fitness value of all nodes is determined.

Step 5: Find the capacity of each node for the process of task allocation.

Step 6: Determine the length of the task, which helps to identify whether the node is suited for the task allocation.

Step 7: The nearest to under utilized nodes are determined for task allocation with the help of the upper and lower bound values.

Step 8: If the underutilized node is determined, then the rand function is used for fixing the upper and lower bounds.

Step 9: Determine the right and left nodes of the underutilized nodes

Step 10: The rand function is used for determining the overloading of the identified underutilized nodes

Step 11: The new underutilized nodes are determined by upper and lower bounds using the rand function.

Step 12: The new underutilized nodes are demined based on the left and right neighbors if the underutilized nodes are overloaded.

Step 13: The limits of the lower and upper bounds are determined.

Step 14: The underutilized nodes' left as well as right nodes are determined.

Step 15: The position of the neighboring nodes is frequently updated for the dynamic task allocation.

Step 16: The position of the nodes is upgraded.

Step 17: The new incoming tasks are preceded to the underutilized nodes.

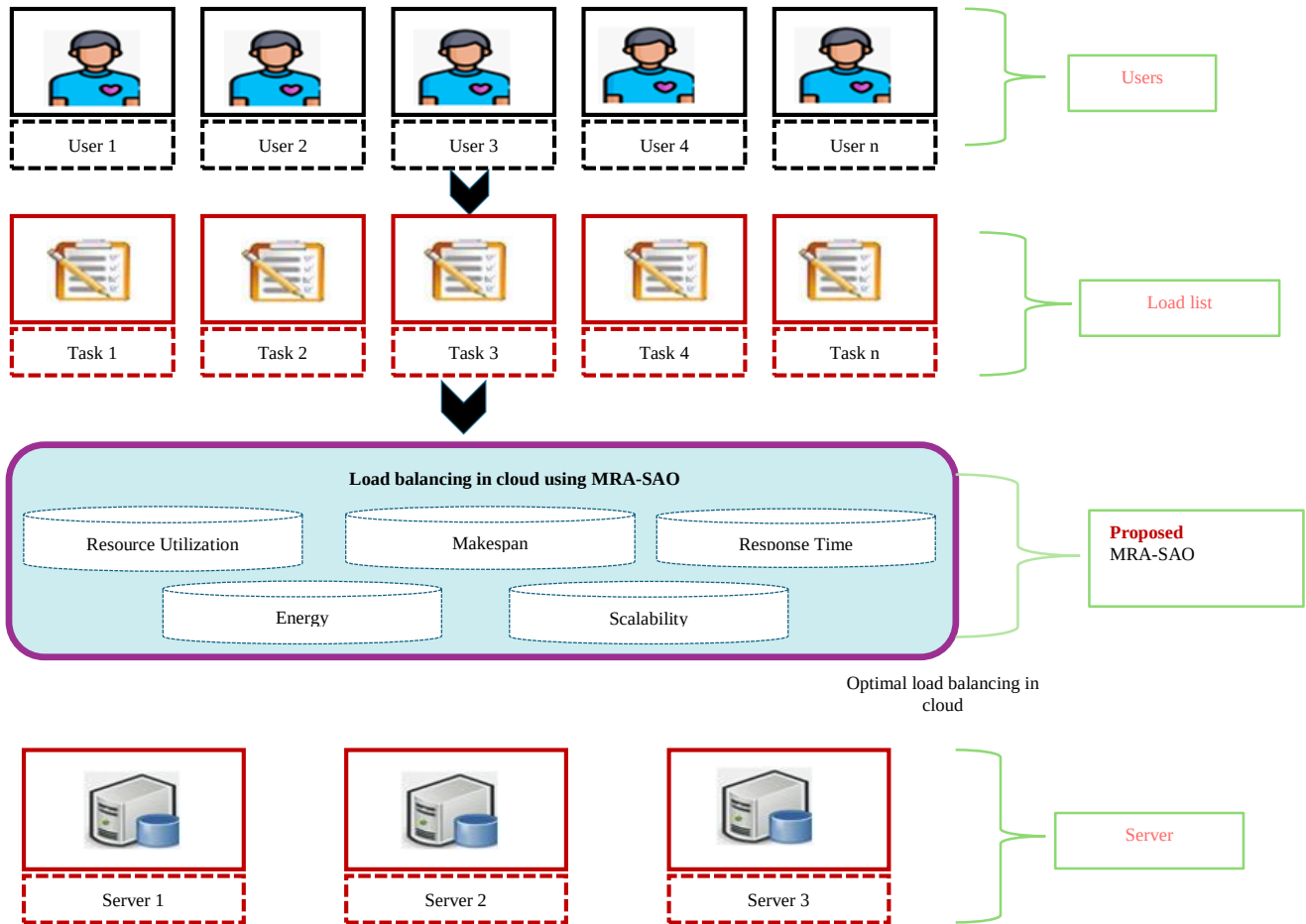


Fig. 3 A pictorial illustration of the MRA-SAO-based load balancing

5.3. Multi-Objective Constraints

In this work, MRA-SAO is introduced to manage workloads across the cloud architecture. Multi-objective constraints are incorporated to enhance cloud load distribution efficacy.

Descriptions of these objective constraints follow, highlighting key factors critical for optimizing network performance.

Resource Utilization RY : Aggregate resources consumed during task execution within the cloud environment, determined via Equation (19).

$$RY = \frac{trc}{trs} \times 100 \quad (19)$$

Here, the total resource capacity is expressed as trc , and the total resource usage is denoted as trs .

Makespan trs : Cumulative duration needed for task execution, computed using Equation (20).

$$MK = \frac{CTT}{STT} \quad (20)$$

Here, Job completion duration is represented by CTT, whereas job initiation time is designated by STT.

Scalability SC : It is the measure of how well the proposed model handles the increased load, and it is expressed in Equation (21).

$$SC = \frac{ri}{pi} \quad (21)$$

Here, the increase in resources is denoted as ri , and the performance improvement is represented as pi .

Response Time RT : It is measured by the difference between the completion and submission time, and it is measured using Equation (22).

$$RT = ct - st \quad (22)$$

Here, the term ct expresses the completion time, and the submission time is represented as st .

Energy Consumption EC : The total energy consumed during the load balancing operation.

Capacity $CAPY$: The capacity of each node in the cloud is determined by this factor.

Length of the task of the node LT : The capability of the nodes for resource allocation is determined by identifying the length of the task for the nodes.

5.3.1. Method Overview and Distinction from Existing Work

The proposed approach presents an integrated framework for predicting workloads as well as multi-objective load balancing of resources in cloud computing systems.

It utilizes the technique called Adaptive Bayesian Belief Network and Ridge Regression (ABBN-RR) for workload forecasting purposes, while at the same time implementing Modified Resource Allocation-Swarm Adaptive Optimization (MRA-SAO) algorithm for allocation of tasks to VMs and load balancing processes.

This algorithm uses a Bayesian belief network for determining the probabilistic dependencies between different workload parameters, while making improvements through Ridge Regression, as the latter prevents the problem of multicollinearity of the former. This research distinguishes itself in various ways from other previous studies. To begin with, most studies focus on load prediction and load balancing

as two separate concepts; however, in this study, the two are considered as one integrated concept.

Another distinctive feature is that the hybrid ABBN-RR approach uses probabilistic reasoning and regression learning, thus increasing the accuracy of predictions than individual models would.

Moreover, the use of the MRA-SAO algorithm is not limited to load balancing only, since it is used to optimize model parameters such as hidden layer count, learning rate, and epoch count.

5.3.2. Dataset Description

The proposed approach for predicting the load with the help of the load balancing model was executed on MATLAB 2020a.

This model contains N instances of the load that describe the patterns of cloud task executions. These instances are characterized by parameters such as CPU usage, memory consumption, task duration, arrival rate of tasks, and processing time.

Before performing the analysis, the data set should be normalized, scaled, and checked for missing values. In addition, the data is partitioned into training (70%), validation (15%), and testing (15%) subsets.

Table 2. Symbols and descriptions of parameters used in the proposed model

Symbol	Description
W_t	Workload at time t
$\hat{W}_t$	Predicted workload
X	Input feature vector
Y	Actual output
$\hat{Y}$	Predicted output
$f(\cdot)$	ABBN-RR prediction function
θ	Model parameters
H	Number of hidden neurons
α	Learning rate
E	Number of epochs
F	Objective (fitness) function
VM_i	i^{th} Virtual Machine
U	Resource utilization
M	Makespan
R_t	Response time
E_c	Energy consumption

6. Results and Discussion

6.1. Experimental Evaluation

The suggested load prediction with the load balancing model was run on the MATLAB2020a software. The well-known heuristic algorithms such as JAYA [29], Moth Flame Optimizer (MFO) [30], Drawer Algorithm (DA) [31], SAO

[26], and existing techniques VMMISD [3], EPRUA [5], and CNN-LSTM [7], served as benchmarks for assessing framework efficacy.

6.2. Performance Metrics

The “L1Norm, L2norm, L_{inf} Norm, Mean Absolute Scaled Error (MASE), Mean Error Percentage (MEP), and Symmetric Mean Absolute Percentage Error (SMAPE) were used for verifying the performance of the developed load prediction model.

$$L1norm = \sum_j |M_j| \quad (23)$$

$$L2norm = \left(\sum_{j=1}^k M_j^2 \right)^{1/2} \quad (24)$$

$$L_{inf} norm = \max_{1 \leq j \leq k} |M_j| \quad (25)$$

$$MASE = mean \left(\frac{|\hat{q}_j|}{\frac{1}{k-1} \sum_{j=1}^k (\hat{q}_j - q_j)} \right) \quad (26)$$

$$MEP = \frac{100 \sum_{j=1}^k |\hat{q}_j - q_j|}{k} \quad (27)$$

$$SMAPE = \frac{100 \sum_{j=1}^k |\hat{q}_j - q_j|}{\frac{|\hat{q}_j| + |q_j|}{2}} \quad (28)$$

6.3. Performance Analysis on Load Prediction Model

The assessment of the proposed MRA-SAO-ABBNRR-based load prediction model is given in Figure 4. The simulation results attained by the proposed MRA-SAO-ABBNRR-model in terms of various activation functions are used for the theoretical explanation of the proposed load prediction model. Referring to Figure 4, the proposed MRA-SAO-ABBNRR-model attains the lower MASE value in the Tanh activation function. A detailed analysis is performed on the proposed MRA-SAO-ABBNRR-based workload prediction model on the diverse activation function interval. In the reported experiment, the proposed MRA-SAO-ABBNRR provides SMAPE of 0.2 in the sigmoid activation function, while the remaining models reach the SMAPE of 1.6, 1, 0.9, and 1.2 for BBN-RR, CNN-LSTM, EPRIJA, VMMISD, respectively. Thus, the results showed that the suggested MRA-SAO-ABBNRR makes accurate results in the load prediction to reduce the operational cost of the cloud system, convincingly; the violation in the cloud system is prevented by the proposed load prediction model. A similar activation function-based comparison is carried out against existing optimization algorithms integrated with the ABBN-RR predictor, namely SAOA-ABBNRR, MFOA-ABBNRR, DA-ABBNRR, and JAYA-ABBNRR, as shown in Figure 5. Across all four activation functions, the proposed MRA-SAO-ABBNRR consistently records the lowest error values for L1Norm, L2Norm, L_{inf} Norm, MAE, MASE, MEP, RMSE, and SMAPE, confirming that the MRA-SAO optimizer contributes to the predictive accuracy of the model independent of the underlying activation function.

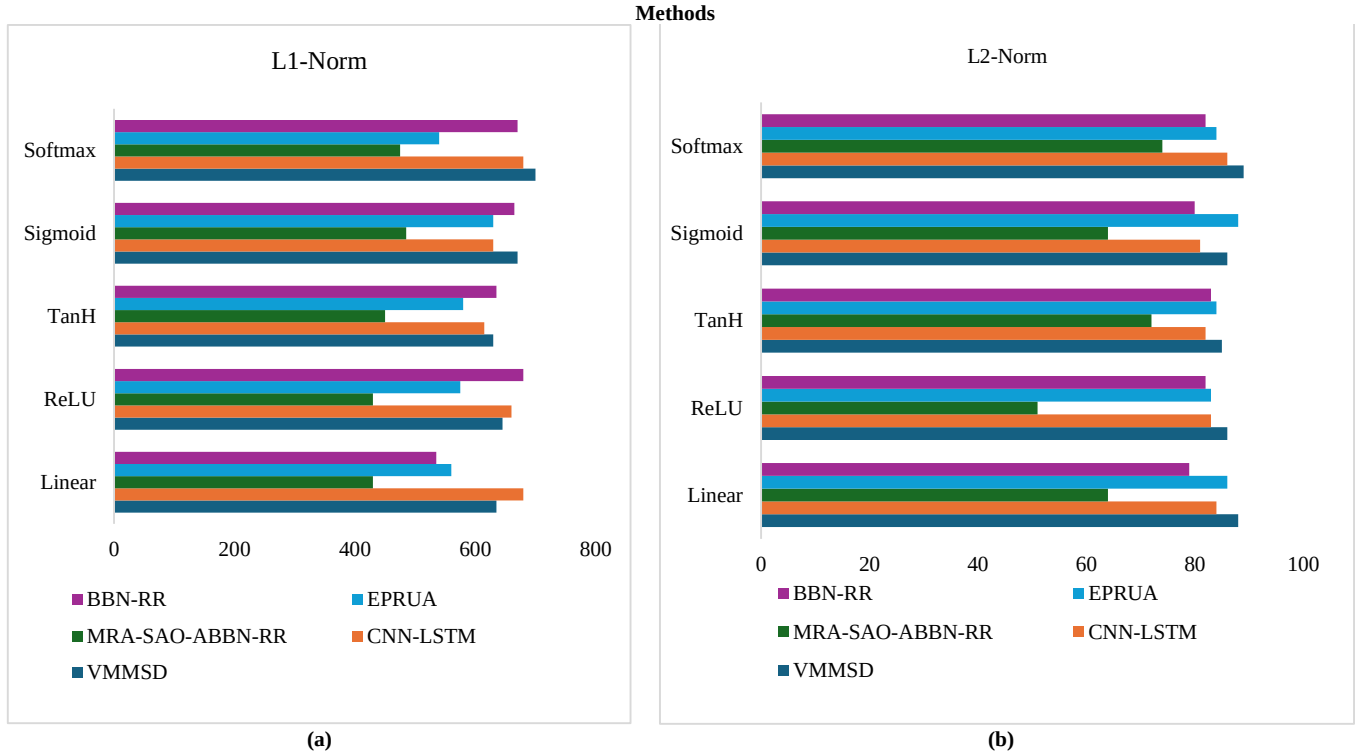
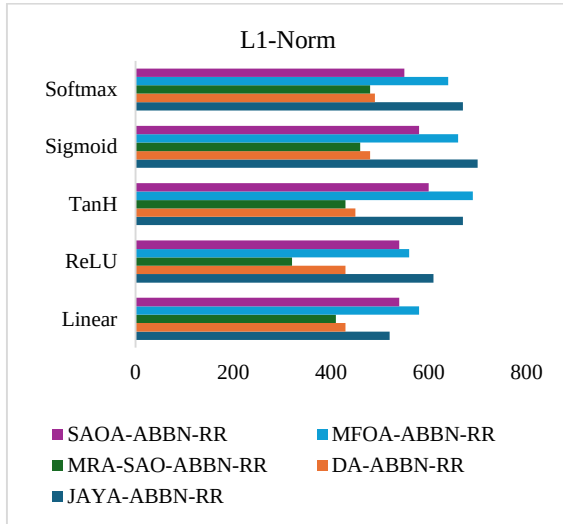


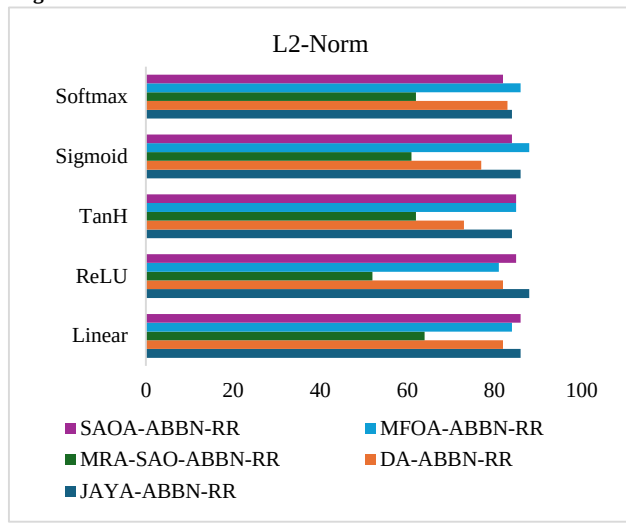


Fig. 4 Activation function-based analysis of the proposed load prediction model against existing prediction methods for (a) L1Norm, (b) L2Norm, (c) L_{inf} Norm, (d) MAE, (e) MASE, (f) MEP, (g) RMSE, and (h) SMAPE.

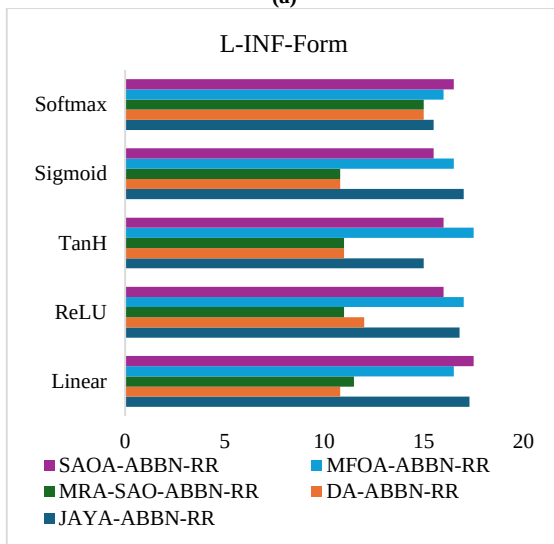
Algorithms



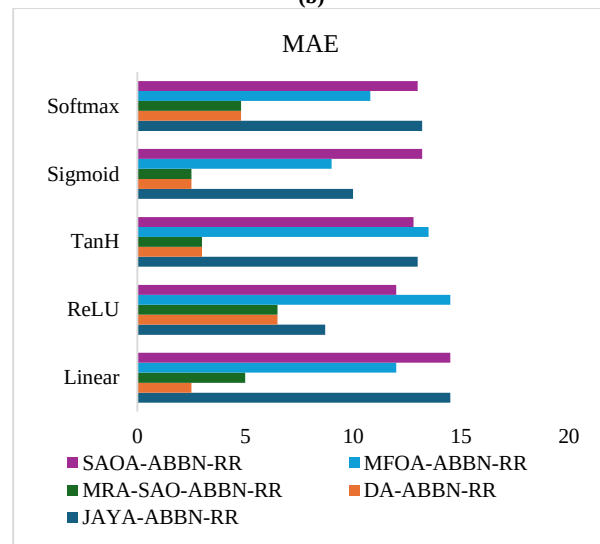
(a)



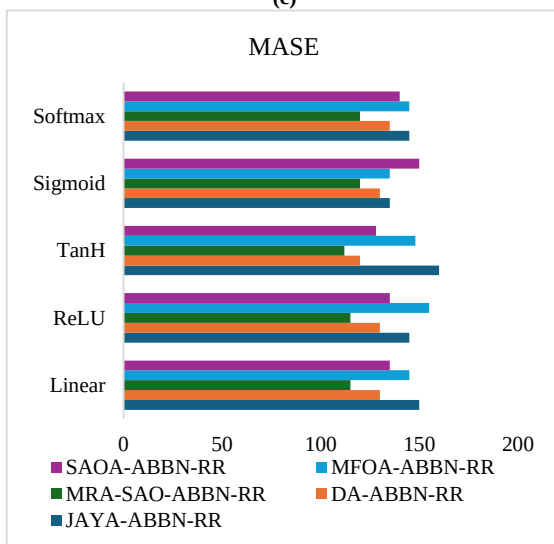
(b)



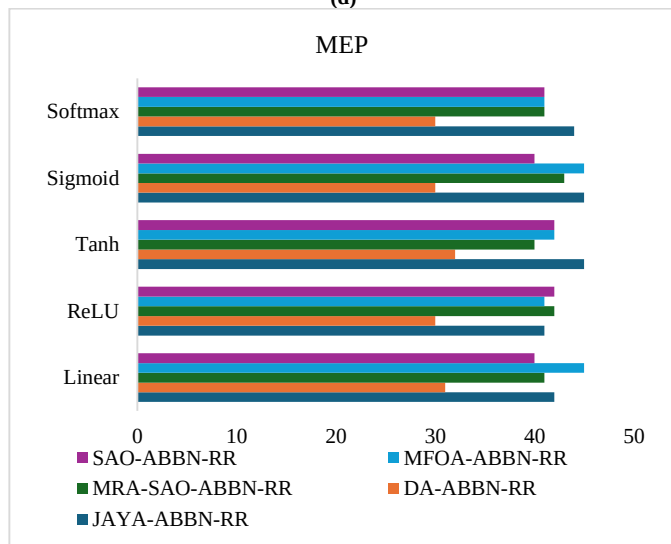
(c)



(d)



(e)



(f)

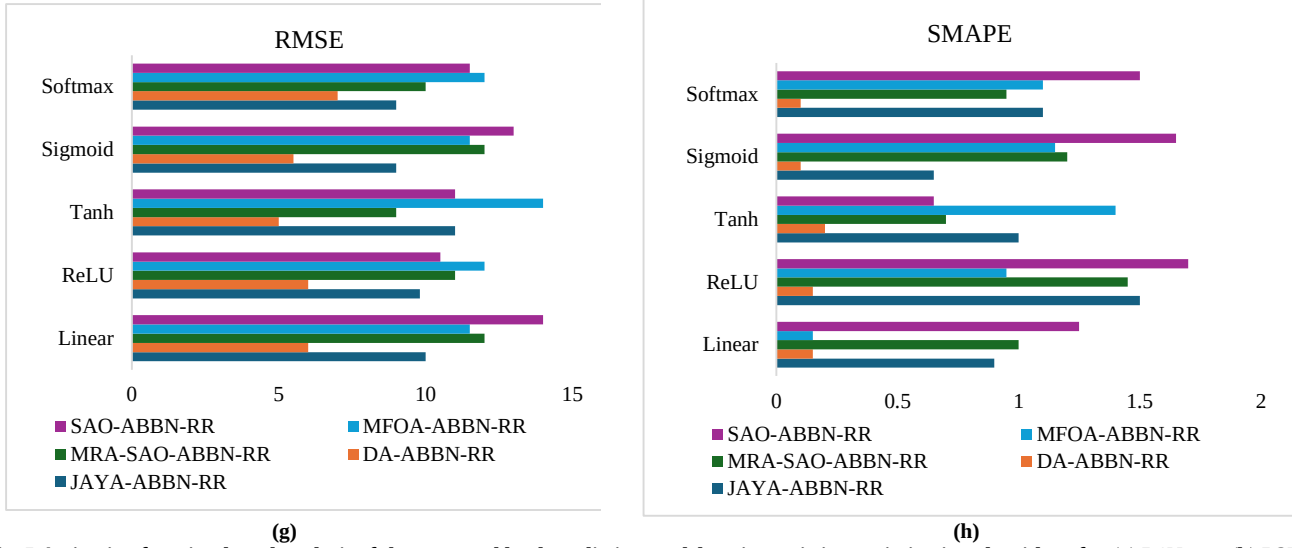
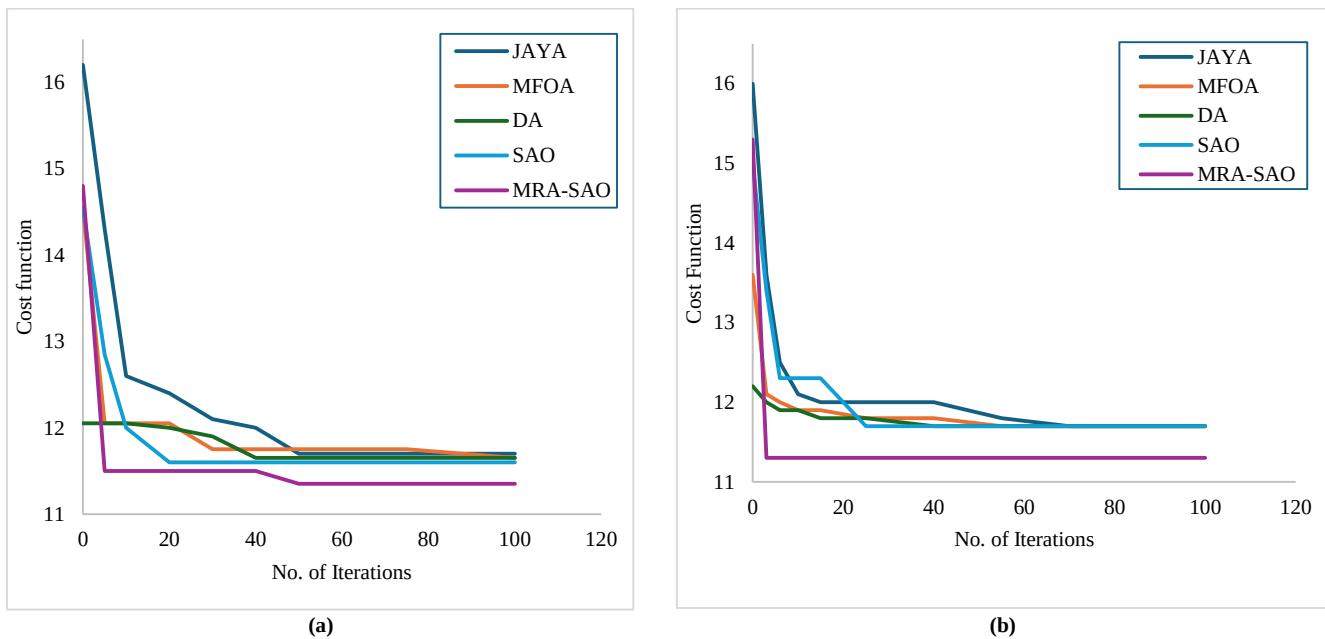


Fig. 5 Activation function-based analysis of the proposed load prediction model against existing optimization algorithms for (a) L1Norm, (b) L2Norm, (c) L_{inf} Norm, (d) MAE, (e) MASE, (f) MEP, (g) RMSE, and (h) SMAPE.

6.4. Convergence Analysis

The performance of the MRA-SAO in the load balancing process is measured by convergence analysis in Figure 6. The convergence assessment is taken on the proposed MRA-SAO with different configurations. The number of servers and tasks in Configuration 1 is 10 and 150, respectively. Likewise, configurations 2, 3, 4, and 5 have a number of servers of 25, 55, 70, and 100, respectively. On the contrary, configurations 2, 3, 4, and 5 have the number of tasks as 250, 300, 400, and 545, respectively. The convergence performance of the MRA-SAO is measured for the other models. In configuration 2, the convergence of the proposed model shows a flat line.

This indicates the developed MRA-SAO showcases the best performance from the initial to the final stage of the iteration without stagnation in the condition of local optima. The results agree that compared to the existing algorithms, such as JAYA, MFOA, DA, and SAO, the designed MRA-SAO provides acceptable converge. Once capability within the minimum iteration values. The rapid convergence capability of the proposed MRA-SAO indicates that it can satisfy the condition of global convergence without falling into the region of local optima. So, the dynamic load balancing on cloud computing is effectively performed with the help of the MRA-SAO.



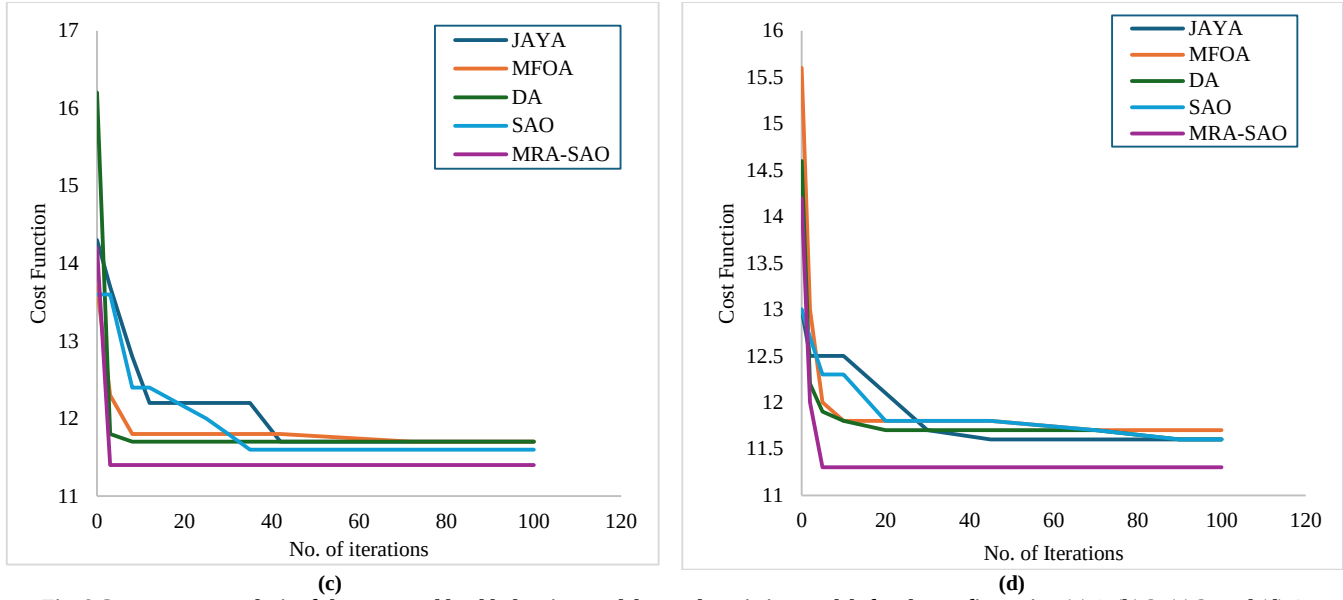
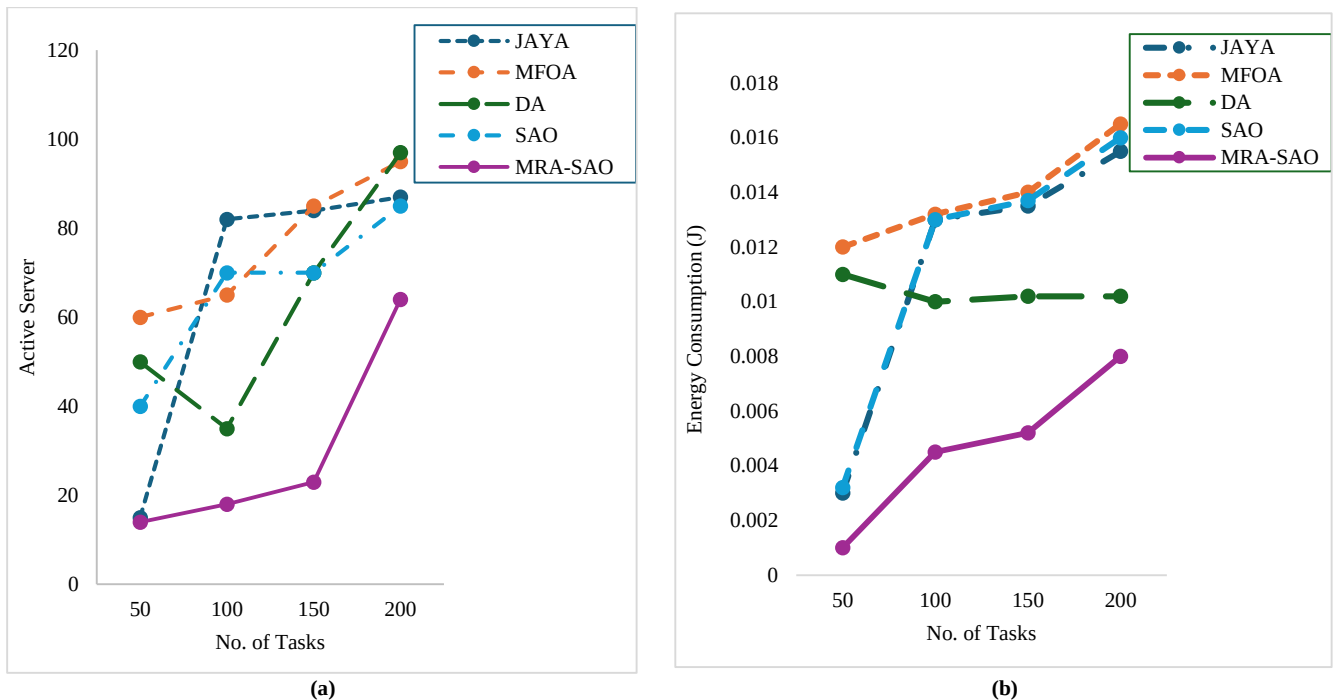


Fig. 6 Convergence analysis of the proposed load balancing model over the existing models for the configuration (a) 1, (b) 2, (c) 3, and (d) 4.

6.5. Analysis of Task Scheduling Model

The results of the proposed MRA-SAO on the load balancing operation in terms of various parameters are visualized in Figure 7. The significant efficiency of the proposed MRA-SAO on the load balancing operation is indicated by the makespan metric since its value is increased with the increase in several tasks in the cloud system. Here, the proposed MRA-SAO looks for some suitable nodes in the cloud for performing the load transmission, and here the time required for the proposed MRA-SAO for transmitting the load is comparatively lower than the JAYA, MFOA, DA, and SAO.

Further, the proposed MRA-SAO showed better performance in the load balancing operation since the proposed MRA-SAO found a suitable server for distributing the load in the cloud system. Compared to the other JAYA, MFOA, DA, and SAO, the proposed MRA-SAO is considered as one of the efficient load-balancing algorithms since it is equipped with features that can determine the overloaded VM with the cloud environment. Thus, the results clearly illustrate that the recommended MRA-SAO provides favorable performance in the load balancing operation and improves the overall efficiency of the cloud system.



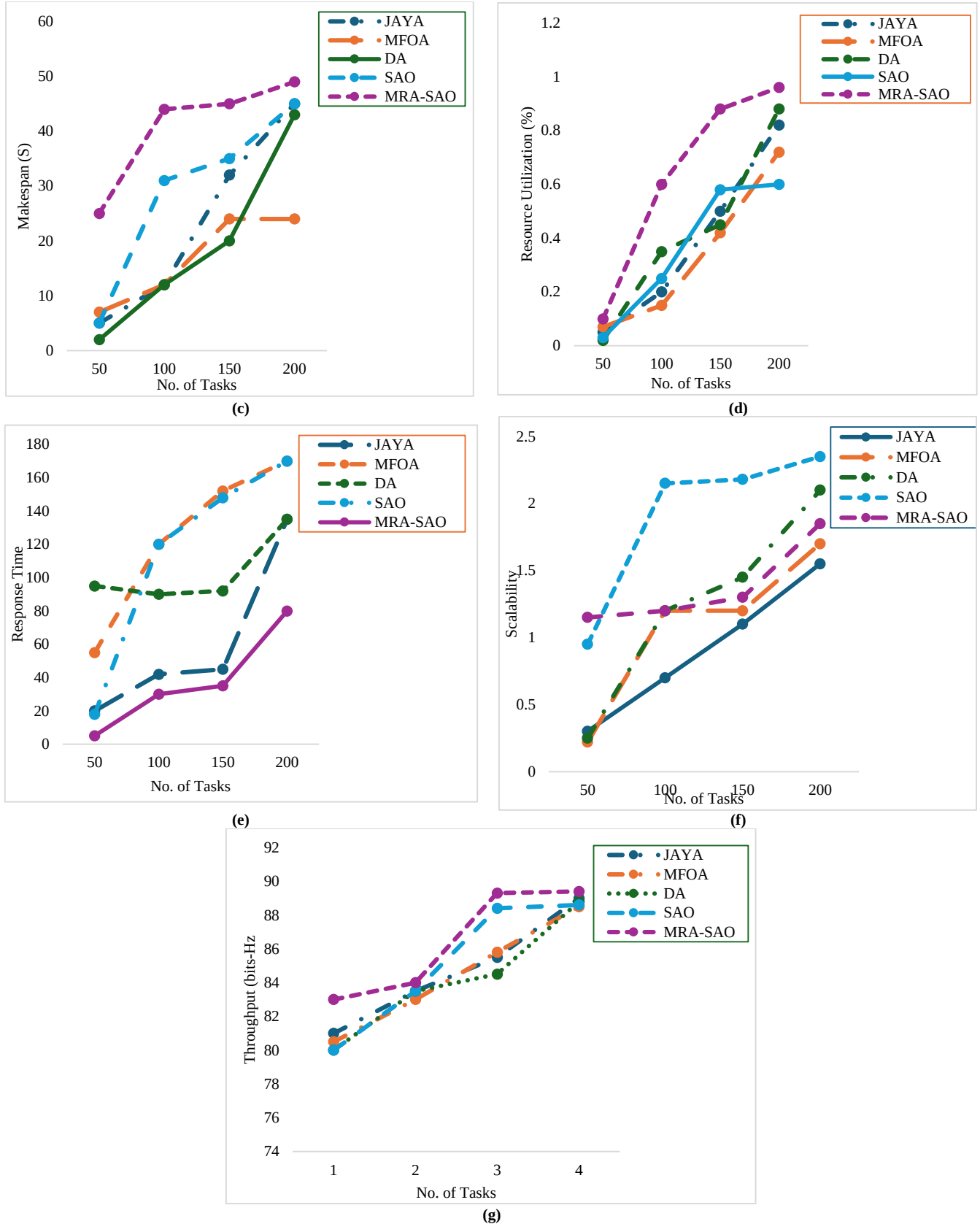


Fig. 7 Performance assessment of the developed load balancing model across (a) Active servers, (b) Energy consumption, (c) Makespan, (d) Resource utilization, (e) Response time, (f) Scalability, and (g) Throughput.

6.6. Comparative Evaluation against Existing Paradigms

Data presented in Table 3 compares the performance of the proposed MRA-SAO framework during load-balancing operations. Results indicate that the introduced MRA-SAO achieves an energy consumption metric of 0.009, substantially lower than existing benchmark approaches. Findings demonstrate that the suggested MRA-SAO successfully balances cloud workloads while delivering accelerated

response times and improved user satisfaction relative to previous models. Resource utilization within the cloud environment scales effectively under the proposed MRA-SAO, ensuring equitable workload distribution across all virtual machines. Consequently, evaluation outcomes confirm that the designed model efficiently resolves cloud load-balancing challenges and enhances overall system productivity.

Table 3. State of art analysis of the proposed load balancing model over the models

Configuration	1	2	3	4	5
Energy Consumption					
VMMISD [3]	0.003121	0.005256	0.007187	0.011304	0.011873
EPRUA [5]	0.007037	0.00777	0.007783	0.009335	0.014406
CNN-LSTM [7]	0.001089	0.007502	0.014856	0.014965	0.017514
BBN-NR	0.001503	0.011682	0.011823	0.014461	0.017713
MRA-SAO	0.000279	0.002397	0.003009	0.008168	0.009484
Makes Span					
VMMISD [3]	1.1354	21.15	15.987	29.722	34.801
EPRUA [5]	4.7205	8.9465	31.93	36.515	43.984
CNN-LSTM [7]	1.6899	15.643	17.2	29.93	40.89
BBN-NR	3.4496	8.0826	13.044	23.552	29.208
MRA-SAO	5.3974	21.269	34.997	45.316	49.133
Average Resource Utilization					
VMMISD [3]	0.28101	0.24071	0.53086	0.7788	0.87537
EPRUA [5]	0.15366	0.44009	0.53133	0.65445	0.94362
CNN-LSTM [7]	0.090823	0.32515	0.40762	0.52714	0.63771
BBN-NR	0.10563	0.26647	0.45742	0.71836	0.95769
MRA-SAO	0.42345	0.51805	0.61096	0.81998	0.96865
Number of Task 200					
VMMISD [3]	32.931	70.851	79.314	80.247	92.439
EPRUA [5]	13.219	30.164	41.002	44.809	70.78
CNN-LSTM [7]	10.604	51.62	70.105	70.463	80.627
BBN-NR	48.191	52.422	64.195	72.563	85.995
MRA-SAO	10.104	16.119	36.016	39.022	51.482

6.6.1. Statistical Validation and Confidence Interval Analysis

To increase the strength and validity of experimental findings, the statistical verification procedure has been included in this research study. Along with the descriptive parameters that indicate the level of model performance, the confidence intervals and statistical testing procedures have been applied.

Firstly, the 95% Confidence Interval (CI) is found for all key metrics that include energy consumption, makespan, resources' efficiency, and time required to accomplish each operation.

Confidence interval reflects the possible deviation from the actual metric value in case if there is some uncertainty. CI can be found by the following formula:

$$CI = \bar{x} \pm Z_{\alpha/2} \cdot \frac{\sigma}{\sqrt{n}} \quad (29)$$

Where $\bar{x}$ indicates the mean value of samples, σ is the standard deviation, n is sample size, while $Z_{\alpha/2}$ represents the critical value corresponding to the chosen confidence level.

Moreover, a statistical significance test with $\alpha=0.05$ will be performed to identify whether there are statistically valid improvements over the baseline solutions. The computed statistical measures, including mean value, standard deviation, and confidence interval for all settings, are presented in Table 3.

6.6.2. Discussion with Supporting and Contradictory Studies

As per the results from the experiments, the proposed ABBN-RR approach combined with MRA-SAO framework performs better compared to other methods in respect of energy efficiency, resource management, and load balancing. The results from the statistical validation, which were provided in the form of Table 3, further confirmed this statement.

Several recent studies suggested that the combination of hybrid techniques and optimization algorithms performed better than the traditional load-balancing algorithms. The authors mentioned that it was mainly because of the prediction and optimization methods used in the former category. It was similar to the approach adopted in the proposed model. But several researchers also claimed that the use of hybrid models and meta-heuristics in cloud computing resulted in more computational overheads. Hence, its application would be difficult for large-scale cloud systems. Besides, the traditional load-balancing algorithm could be sufficient if applied in a relatively stable environment. In addition, it was observed that several studies that focused only on performance metrics did not provide any measure of uncertainty, such as confidence intervals and statistical tests of significance. Hence, the results from such studies might be difficult to reproduce or validate.

7. Conclusion

In recent years, workload distribution has become pivotal for improving cloud reliability. Consequently, this study introduced an innovative framework for cloud workload forecasting and balancing. Initially, the ABBNRR was proposed for predicting the load within the cloud system. Further, the load balancing operation was done by the MRA-SAO algorithm, where the load was balanced to the VM based on the objective constraints. In the reported experiment, the proposed MRA-SAO-ABBNRR provides SMAPE of 0.2 in the sigmoid activation function, while the remaining models reach the SMAPE of 1.6, 1, 0.9, and 1.2 for BBN-RR, CNN-LSTM, EPRIJA, VMMISD, respectively. Performance comparisons confirm that the introduced architecture outperforms existing baseline models. Future research directions will focus on incorporating virtual machine migration strategies driven by cloud server workload dynamics. Furthermore, additional workload characteristics will be explored in subsequent studies to further refine the efficiency of the developed system.

7.1. Limitations

However, there are several limitations that must be pointed out despite the impressive results of the developed ABBN-RR with MRA-SAO solution. Firstly, it involves the use of hybrid predictive and optimizing models, thus raising concerns about its computational complexity and the ability to scale well for very high load rates in cloud computing environments. Secondly, the model is dependent on workload data in the past; hence, it may underperform in case of drastic and irregular workloads. Thirdly, the testing of the approach was done in a simulation environment rather than in an actual cloud environment with its inherent heterogeneity.

References

- [1] Arabinda Pradhan et al., "Intelligent Decision-Making of Load Balancing using Deep Reinforcement Learning and Parallel PSO in Cloud Environment," *IEEE Access*, vol. 10, pp. 76939-76952, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [2] Mana Saleh Al Reshan et al., "A Fast Converging and Globally Optimized Approach for Load Balancing in Cloud Computing," *IEEE Access*, vol. 11, pp. 11390-11404, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

7.2. Practical Implications

Practically speaking, the model suggested in this work will be of considerable importance for contemporary cloud computing architectures. Using the prediction of workload with a multi-objective load balancing approach helps in efficient resource management and minimizes energy consumption, as well as enhances task processing speed. Such advantages of the model will be especially important for large-scale computing facilities, since energy efficiency affects the operational costs and sustainability of the operations. The possibility of adaptation of the model to changing workloads makes it applicable for real-time data processing tasks, IoT applications, and high-performance computing infrastructure.

7.3. Future Research Directions

Several potential research avenues could be pursued to further develop the presented framework. First, it is possible to modify and expand the model to make it feasible for application within the scope of real-time deployment within distributed or edge-cloud systems, which is highly dependent on low-latency requirements. Second, one can consider the use of deep neural networks, particularly transformer-based ones, in order to further boost the workload prediction precision. Another possibility is to include an adaptive and self-optimizing component that would minimize computations. Moreover, multi-cloud/federated cloud scenarios could be considered in the context of increased complexity associated with resource allocation processes within such environments.

Conflicts of Interest

The authors declare the absence of any known financial ties, personal connections, or competing affiliations that might have influenced the research presented in this work.

Funding Statement

This study received no external financial support from public, commercial, or non-profit funding organizations.

Acknowledgements

The authors gratefully acknowledge the Department of Computer Science and Engineering at GITAM (Deemed to be University), Hyderabad, and the Department of Computer Science and Engineering (Data Science) at CVR College of Engineering, Hyderabad, for providing the necessary academic resources and research facilities. Appreciation is also extended to the anonymous reviewers for their constructive feedback and recommendations, which significantly enhanced the quality of this manuscript.

- [3] Madala Guru Brahmam, and Vijay Anand R, "VMMISD: An Efficient Load Balancing Model for Virtual Machine Migrations via Fused Metaheuristics with Iterative Security Measures and Deep Learning Optimizations," *IEEE Access*, vol. 12, pp. 39351-39374, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [4] Kaviarasan R et al., "Effective Load Balancing Approach in Cloud Computing using Inspired Lion Optimization Algorithm," *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, vol. 6, pp. 1-12, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [5] P. Udayasankaran, and S. John Justin Thangaraj, "Energy Efficient Resource Utilization and Load Balancing in Virtual Machines using Prediction Algorithms," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 127-134, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [6] B. Edward Gerald, P. Geetha, and E. Ramaraj, "A Fruitfly-based Optimal Resource Sharing and Load Balancing for the Better Cloud Services," *Soft Computing*, vol. 27, no. 10, pp. 6507-6520, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [7] Sarita Simaiya et al., "A Hybrid Cloud Load Balancing and host Utilization Prediction Method using Deep Learning and Optimization Techniques," *Scientific Reports*, vol. 14, no. 1, pp. 1-18, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [8] Amanpreet Kaur et al., "Load Balancing Optimization based on Deep Learning Approach in Cloud Environment," *International Journal of Information Technology and Computer Science(IJITCS)*, vol. 12, no. 3, pp. 8-18, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [9] G. Muneeswari et al., "GEP Optimization for Load Balancing of Virtual Machines (LBVM) in Cloud Computing," *Measurement: Sensors*, vol. 33, pp. 1-6, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [10] Deepal Tennakoon, Morshed Chowdhury, and Tom H. Luan, "Cloud-based Load Balancing using Double Q-Learning for Improved Quality of Service," *Wireless Networks*, vol. 29, no. 3, pp. 1043-1050, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [11] Jyoti Singh et al., "Load-Balancing Strategy: Employing a Capsule Algorithm for Cutting Down Energy Consumption in Cloud Data Centers for Next Generation Wireless Systems," *Computational Intelligence and Neuroscience*, vol. 2023, no. 1, pp. 1-13, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [12] Maria Jesi et al., "Load Balancing in Cloud Computing via Mayfly Optimization Algorithm," *Romanian Journal of Technical Sciences – Electrical Engineering and Energy*, vol. 69, no. 1, pp. 79-84, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [13] Adeyinka Ajao Adewale et al., "Hybridization of the Q-Learning and Honey Bee Foraging Algorithms for Load Balancing in Cloud Environments," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 4, pp. 4602-4615, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [14] Kavish Chawla, "Reinforcement Learning-based Adaptive Load Balancing for Dynamic Cloud Environments," *arXiv preprint*, pp. 1-6, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [15] Prathamesh Vijay Lahande et al., "Reinforcement Learning Approach for Optimizing Cloud Resource Utilization with Load Balancing," *IEEE Access*, vol. 11, pp. 127567-127577, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [16] S. Sree Priya, and T. Rajendran, "Improved Round-Robin Rule Learning for Optimal Load Balancing in Distributed Cloud Systems," *International Journal of System of Systems Engineering*, vol. 13, no. 1, pp. 83-99, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [17] Minal Shahakar, Surenda Mahajan, and Lalit Patil, "Load Balancing in Distributed Cloud Computing: A Reinforcement Learning Algorithms in Heterogeneous Environment," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 2, pp. 65-74, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [18] Ankita Srivastava, and Narander Kumar, "An Efficient Firefly and Honeybee based Load Balancing Mechanism in Cloud Infrastructure," *Cluster Computing*, vol. 27, no. 3, pp. 2805-2827, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [19] Thanwamas Kassanuk, and Khongdet Phasinam, "A Hybrid Binary Bird Swarm Optimization (BSO) and Dragonfly Algorithm (DA) for VM Allocation and Load Balancing in Cloud," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 13, no. 1, pp. 1-21, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [20] Raghavender Reddy Kothi Laxman et al., "PGWO-AVS-RDA: An Intelligent Optimization and Clustering based Load Balancing Model in Cloud," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 21, pp. 1-10, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [21] Garima Verma, "Load Balancing in Cloud Environment using Opposition based Spider Monkey Optimization," *Wireless Personal Communications*, vol. 137, no. 2, pp. 977-996, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [22] Shruthi G. et al., "Deep Learning-based Resource Prediction and Mutated Leader Algorithm Enabled Load Balancing in Fog Computing," *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 15, no. 4, pp. 84-95, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [23] Peng Yang et al., "Reducing Idleness in Financial Cloud Services Via Multi-Objective Evolutionary Reinforcement Learning based Load Balancer," *Science China Information Sciences*, vol. 67, no. 2, pp. 1-21, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [24] Tanzila Saba et al., "Cloud-Edge Load Balancing Distributed Protocol for IoE Services using Swarm Intelligence," *Cluster Computing*, vol. 26, no. 5, pp. 2921-2931, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [25] Ibrahim Aqeel et al., "Load Balancing using Artificial Intelligence for Cloud-Enabled Internet of Everything in Healthcare Domain," *Sensors*, vol. 23, no. 11, pp. 1-15, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [26] Lingyun Deng, and Sanyang Liu, "Snow Ablation Optimizer: A Novel Metaheuristic Technique for Numerical Optimization and Engineering Design," *Expert Systems with Applications*, vol. 225, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [27] Vivek Shrivastava, and R.B. Misra, "Development of Bayesian Belief Network Model for Electrical Load Demand," *International Journal of Systems Assurance Engineering and Management* vol. 1, no. 2, pp. 170-177, 2010. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [28] Jae Joon Ahn et al., "Using Ridge Regression with Genetic Algorithm to Enhance Real Estate Appraisal Forecasting," *Expert Systems with Applications*, vol. 39, no. 9, pp. 8369-8379, 2012. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [29] Zhiyu Feng et al., "A Jaya Algorithm based on Self-Adaptive Method for Parameters Identification of Photovoltaic Cell and Module," *Cluster Computing*, vol. 28, no. 2, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [30] Linfei Yin, and Wenyu Ding, "Dual Deep Neural Networks-Accelerated Non-Dominated Sorting Moth Flame Optimizer for Distributed Multi-Objective Economic Dispatch," *Expert Systems with Applications*, vol. 259, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [31] Eva Trojovská, Mohammad Dehghani, and Víctor Leiva, "Drawer Algorithm: A New Metaheuristic Approach for Solving Optimization Problems in Engineering," *Biomimetics*, vol. 8, no. 2, pp. 1-35, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]