I. J. Computer Network and Information Security, **2026, 4, 187-209**
Published Online on August 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2026.04.10

Enhancing Energy Efficiency in Cloud Data Centers through Deformable Graph Convolutional Network-Aware Virtual Machine Placement with Hybrid Swarm Bipolar Walk-Spread Optimization

Viji Vinod*
Faculty of Computer Applications, Dr. M. G. R. Educational and Research Institute, Maduravoyal, Chennai, Tamil Nadu 600095, India
E-mail: hod-mca@drmgrdu.ac.in
ORCID iD: https://orcid.org/0000-0002-1259-1958
*Corresponding author

V. J. Chakravarthy
Faculty of Computer Applications, Dr. M. G. R. Educational and Research Institute, Maduravoyal, Chennai, Tamil Nadu 600095, India
E-mail: chakravarthy.mca@drmgrdu.ac.in
ORCID iD: https://orcid.org/0000-0003-2139-0709

N. Jayashri
Faculty of Computer Applications, Dr. M. G. R. Educational and Research Institute, Maduravoyal, Chennai, Tamil Nadu 600095, India
E-mail: jayashri.mca@drmgrdu.ac.in
ORCID iD: https://orcid.org/0000-0003-4314-7638

Bala Dhandayuthapani V.
Department of Information Technology, College of Computing and Information Sciences, University of Technology and Applied Sciences, Shinas campus, Oman
E-mail: bala.veerasamy@utas.edu.om, dhanssoft@gmail.com
ORCID iD: https://orcid.org/0000-0002-8310-0642

Shabeen Taj G. A.
Department of Computer Science Engineering, Government Engineering College Ramanagar, B.M. Road, Ramanagara, Karnataka 562159, India
E-mail: shab2en@gmail.com
ORCID iD: https://orcid.org/0000-0003-2498-4369

A. V. G. A. Marthanda
Department of Electrical and Electronics Engineering, Lakireddy Balireddy College of Engineering, Mylavaram, Andhra Pradesh, India
E-mail: ayyala4u@gmail.com
ORCID iD: https://orcid.org/0000-0003-1085-9620

Received: March 15, 2025; Revised: July 20, 2025; Accepted: September 15, 2025; Published: August 8, 2026

Abstract: In big cloud data centers, the best physical machine (PM) is selected using the Virtual Machine Placement (VMP) process. To address this issue, a number of approaches have been proposed. Nevertheless, the existing solutions only take into account a small number of resource categories, which leads to an uneven load and ultimately, the activation of superfluous physical computers within the data center. The aim of this research is to maximize resource usage while lowering power use and carbon footprints by integrating a Hybrid Swarm Bipolar with Walk-Spread Algorithm with a unique Deformable Graph Convolutional Network (DGCN-SB-WSA)-aware virtual machine

placement architecture. To ensure efficient VM scheduling and management, real-world cloud workloads are analyzed using the Google Cluster Dataset (GCD). The Deformable Graph Convolutional Network (DGCN) dynamically models cloud infrastructure as a graph that captures intricate relationships among PMs and VMs, enabling adaptive placement choices. The Hybrid Swarm Bipolar with Walk-Spread Algorithm (SB-WSA) then uses a dual-phase search approach to balance local exploitation with global exploration, minimizing premature convergence and increasing performance while optimizing Virtual machine (VM) allocation. Comparing the proposed method to conventional VM placement techniques, experimental results show that it dramatically lowers power consumption to 27 kW, improves resource usage to 98%, and decreases carbon emissions to 180 kg CO_2.

Index Terms: Deformable Graph Convolutional Network, Google Cluster Dataset, Swarm Bipolar Optimization, Virtual Machine Placement and Walk-Spread Algorithm.

1. Introduction

Cloud computing has been offering a range of online services in recent years. Cloud platforms are increasingly being used by data centers to implement a wide range of applications [1]. Cloud users can access a variety of services from data centers, which are essential infrastructures. To meet the ever-growing and changing demand of cloud consumers, it is necessary to achieve the Quality of Service (QoS) scale between cloud customers and their service providers [2]. To provide the required QoS for housed applications, cloud service providers need synchronization between resources and power distribution and heat emission inside the data center [3]. In a data center, the placement and maintenance of Virtual Machines determine its efficiency. The VMP has a great influence on the overall operating cost, utilization of power, and occupancy of resources in the data center [4]. The inefficient use of these resources is the main cause for the wastage of energy. Maximum power consumption of cloud data centers increases the operational cost which indirectly increases the level of carbon dioxide CO2 [5]. It has been observed that computers and cooling systems together utilize more than 70% of the total power consumption in a data center [6]. Resource utilization, power consumption, and carbon emission are some of the important parameters used by cloud service providers for decreasing the overall operational cost [7]. Carbon emissions in data centers can be decreased by efficiently allocating server power consumption to a particular workload [8]. There are numerous VM consolidation methods at one's disposal. VM consolidation methods are used with a hardware virtualization mechanism to run the VMs on the least amount of physical machines (PMs) [9]. To ensure lower energy usage and maximize server deployment, the data center's active PMs must be kept to a minimum. The VMP problem has been the subject of numerous studies in an effort to address issues with energy and resource efficiency [10]. The multiobjective VMP problem is therefore a difficult and complex task that falls under the Nondeterministic Polynomial (NP)-complete complexity class [11]. Many studies concentrate on VM consolidation difficulties from the standpoint of cloud service providers in order to address concerns related to power-efficient resource utilization [12]. Using the idea of meta-heuristic optimization approaches, a VMP methodology was introduced in [13]. Overload and underload resource utilization issues are addressed in this paper. With fewer active PMs, the entire workload is aggregated [13]. Additionally, the authors discuss a secure VMP that takes security risks and communication costs into account for applications that require a lot of data. In [14], a VMP technique is discussed that assigns interdependent VMs to PMs that are closer together in order to optimize network traffic and bandwidth characteristics [14]. To solve the problems of cloud users and service providers, the VMP community must consistently work toward each goal at the same time. It calls for a multi-efficient VMP technique that makes use of the prevalent multiplexing and resource-sharing mechanisms [15].

1.1. Problem Statement and Motivation

The challenges are that effective VMP in cloud data centers is essential for lowering operating expenses, cutting power consumption, and guaranteeing optimal resource usage while preserving quality of service. Inefficient VMP results in high energy use, high carbon emission, and resource waste. Because VMP problem is NP-complete, cloud computing systems need an efficient multi-objective technique for VMP that can effectively balance between cost optimization, energy efficiency, and resource allocation [16-23].

1.2. Novelty and Contributions

The proposed Hybrid Swarm Bipolar with Walk-Spread Algorithm (SB-WSA) integrates swarm intelligence and a bipolar decision model to maintain the balance between exploration and exploitation in cloud resource management. Accordingly, the walk-spread mechanism in the algorithm effectively maintains load balancing, reduces resource wastage, and avoids the prematurity of convergence. This provides better CPU and memory utilization with reduced power consumption, carbon emission, and active servers. In dynamic cloud environments, this will lead to enhanced scalability, reduced response time, and improved QoS.

The key contributions of this research are summarized as follows:

- It proposes an effective and efficient approach for the adaptive placement of VMs using the Google Cluster Dataset, which reduces the number of active PMs, energy consumption, and improves workload balance.
- DGCN enhances the VM placement decisions by dynamically modeling relationships between VMs and PMs as a graph structure.
- The Hybrid Swarm Bipolar Walk-Spread Algorithm technique will help to avoid the defect of premature convergence and will ensure optimal resource allocation as it maintains the balance between exploration and exploitation.
- The integrated DGCN–SB-WSA framework minimizes network transmission latency, power usage, and resource wastage by optimizing VM distribution in data centers.
- The performance of the proposed algorithm is evaluated through ANOVA and DMRT statistical analyses to validate the reliability and significance of the results.

The relevant literature is carefully assessed in Section 2. The methods employed in this inquiry are described in detail in Section 3. The findings and the consequences are examined in Section 4. Individual viewpoints and recommendations for additional research are included in Section 5's conclusion.

2. Literature Review

The most recent research on Cloud Virtual Machine Placement is included below.

In 2023, Singh, A.K., et al., [16] have established a bio-inspired virtual machine placement toward sustainable cloud resource management. In this paper, an algorithm called Flower Pollination based Non-dominated Sorting Optimization (FPNSO) is presented it minimizes energy consumption and maximizes resource utilization, thereby lowering carbon emissions in the data center. The VMP is implemented by combining the concepts of Flower Pollination Optimization (FPO) and the Non-dominated Sorting technique based Genetic Algorithm (NSGA-II).

In 2022, Karmakar, K., et al., [17] have introduced the Utilization aware and network I/O intensive virtual machine placement policies for cloud data center. Here, the idea of a virtual cluster is presented, which is composed of virtual machines (VMs) that communicate with one another. To lower the cost of communication, it is preferable to locate the VMs in a cluster adjacent to one another. Additionally, introduced meta-heuristics based on genetic algorithms and algorithms based on semi-definite programming (SDP). Present the idea of a virtual cluster, which is made up of VMs that communicate with one another.

In 2025, Swain, S.R., et al., [18] have developed an intelligent virtual machine allocation optimization model for energy-efficient and reliable cloud environment. This research presents a new VM deployment technique based on resource forecasting that significantly reduces energy consumption and increases system reliability. Here, a self-adaptive differential evolution method is developed for feed-forward neural network optimization that combines global exploration with multidimensional learning. This approach, in contrast to conventional gradient descent algorithms, looks for the global optimal solution, providing more reliable and precise forecasts of future resource consumption.

In 2022, Saxena, D. and Singh, A.K., [19] have established a high availability management model based on VM significance ranking and resource estimation for cloud applications. The present paper introduces a new High Availability Management (SRE-HM) model based on VM Significance Ranking and Resource Estimation. An importance ranking variable was established and computed for every virtual machine (VM). Following the execution of either critical or non-critical activities, an admissible High Availability (HA) strategy is selected in accordance with the task's importance and user-specified limitations.

In 2022, Aghasi, A., et al., [20] have developed a thermal-aware energy-efficient virtual machine placement algorithm based on fuzzy controlled binary gravitational search algorithm (FC-BGSA). Here, one can offer a metaheuristic approach based on the binary version of the gravitational search algorithm to reduce the processing and cooling energy in the Virtual Machine Placement (VMP) problem. Furthermore, propose a fuzzy logic-based self-adaptive strategy to regulate the algorithms' exploration and exploitation behavior.

In 2024, Swain, S.R., et al., [21] have presented an efficient straggler task management in cloud environment using stochastic gradient descent with momentum learning-driven neural networks. In order to assess and classify a variety of occupations into those that are stragglers and those which are not, a neural network-based resource predictor is first developed and modified using the Stochastic Gradient Descent with Momentum approach. Once the straggler tasks have been identified, they are further divided into two groups according to their particular resource needs: Resource Hunters and Long-Tail Stragglers. To improve sustainability and accomplish parallelism in the cloud architecture, a task management policy is put into place.

In 2024, Singh, G., et al., [22] have developed a feature extraction and time warping based neural expansion architecture for cloud resource usage forecasting. The paper offers a hybrid approach that predicts the machine-level workload in a less computationally costly way by combining deep neural learning, transfer learning, and cluster analysis. The method looks for similarities between the input dataset's non-linear computation profiles by using clustering. Once each cluster has been found, a sample dataset is used to create generalized deep neural learning models. On the Google Cluster's real-world traces dataset, the presented method's performance is validated.

In 2025, Kumar, J., et al., [23] have introduced a hybrid neural network and cooperative PSO model for dynamic cloud workloads prediction. Here, a neural network model is provided for workload forecasting in cloud environments that is facilitated by cooperative learning. Cooperative learning guarantees that no important information is lost during optimization, while the model breaks down workload traces into discrete parts so that the neural network can learn patterns from each. The overview of the current research on Cloud Virtual Machine Placement is described in table 1.

Table 1. Overview of Current Research on Cloud VMP Mechanisms.

Reference	Methods	Advantages	Limitations
[16]	FPO-NSGA-II	The capacity to optimize resource use, guaranteeing the effective operation of physical machines (PMs) and lowering the quantity of idle and underutilized servers.	Its computational complexity, which could lead to longer convergence times for large-scale data centres because the combination of FPO and NSGA-II increases the algorithm's processing overhead.
[17]	SDP	Genetic Algorithms (GA) are used to improve performance by iteratively improving placement choices, which lowers the cost of communication between VMs and the use of physical resources.	Despite being more scalable, heuristic-based approaches can occasionally yield less-than-ideal outcomes, particularly when handling workloads that are extremely dynamic.
[18]	FFNN	This method guarantees global exploration, which results in more accurate and reliable resource predictions than classic gradient descent techniques, which can become trapped in local minima.	The balance between system stability and VM consolidation frequency; frequent VM migrations may still affect performance and service quality even though fewer active PMs use less energy.
[19]	SRE-HM	This makes HM more cost-effective than typical approaches that apply redundancy across all VMs by avoiding needless resource allocation for less important VMs.	When predicting resource utilization, the model does not properly account for VM cooperation relationships, which can lead to ineffective migration choices and increased delay in high-availability schemes.
[20]	FC-BGSA	The method successfully strikes a balance between exploration and exploitation by combining gravitational search optimization with a self-adaptive fuzzy logic mechanism.	The algorithm's processing time can be prolonged by the computational overhead brought about by gravitational search optimization, particularly in large-scale cloud systems.
[21]	SGD-ML-DNN	The division of stragglers into Resource Hunters and Long-Tail stragglers enables focused resource distribution, guaranteeing that jobs with particular needs are given the right amount of processing power.	In large-scale cloud systems, the intricacy of training and fine-tuning the neural network-based predictor may result in increased processing overhead and lengthier model training durations.
[22]	DNN-TL	The method dramatically lowers computational cost while preserving good prediction accuracy when applied to extended deep neural learning models that simply use a sample dataset from each detected cluster.	Although transfer learning lowers computing costs, it does not always translate well to highly dynamic cloud systems where workload characteristics are constantly changing and to unforeseen demand patterns.
[23]	HNN-PSO	By guaranteeing that no important information is overlooked, the cooperative learning mechanism maximizes the learning process and increases forecasting accuracy.	Computational complexity brought about by the cooperative learning mechanism and the breakdown of workload traces, which could lengthen training times and use more resources.

A closer examination of recent GCN-based and hybrid optimization-driven VM placement models reveals distinct methodological gaps that motivate the proposed DGCN–SB-WSA approach. Traditional GCN-based approaches [22,24] effectively model the spatial and relational dependencies among VMs and PMs, but they often suffer from static graph structures, hindering their adaptability under dynamic workloads. On the other hand, hybrid metaheuristic schemes, such as FPO-NSGA-II [16] and FC-BGSA [20], enhance energy efficiency with relatively slower convergence and sensitivity to initial parameter settings. Few works incorporate adaptive graph learning with dual-phase swarm optimization that can achieve both global search capability and structural flexibility. The proposed DGCN–SB-WSA framework overcomes the aforementioned limitations by combining a deformable graph representation for dynamic dependency modeling with a hybrid Swarm Bipolar–Walk-Spread optimizer to enhance the exploration–exploitation balance. The synergy allows the derivation of more accurate and energy-aware VM placement compared with existing single-domain methods.

2.1. Challenges

Efficient VMP is considered indispensable for cloud data centers in order to save costs, bound power usage, and maximize resource utilization. In contrast, inefficient VMP leads to the underutilization of resources, high energy consumption, and carbon emission. Due to its NP-completeness, the existing solutions have scalability issues, dynamic

workloads, and computational complexity. System dependability and quality of service need a multi-objective VMP strategy that balances cost optimization, resource allocation, and energy efficiency.

3. Proposed Virtual Machine Placement Methodology

The proposed methodology combines a Hybrid Swarm Bipolar with Walk-Spread Algorithm and a Deformable Graph Convolutional Network to obtain the optimal placement of virtual machines for energy efficiency in cloud data centers. This technique considers the cloud infrastructure as a graph, where virtual machines and physical machines act as nodes, while their dependencies act as edges. On the basis of system constraints and resource requirements, the DGCN framework changes its graph topology dynamically to maximize the placement of virtual machines. After VM allocation, the SB-WSA refines the placement in two optimization phases: the swarm bipolar mechanism, which balances attraction and repulsion forces to avoid premature convergence, and the walk-spread approach for the exploration of both global and local search spaces to fine-tune resource utilization. By reducing the power consumption, carbon emission, and wastage of resources, this framework enhances energy efficiency and assures better performance in the cloud data centers with a balanced distribution of workload. The proposed system's architecture is shown in Fig. 1.

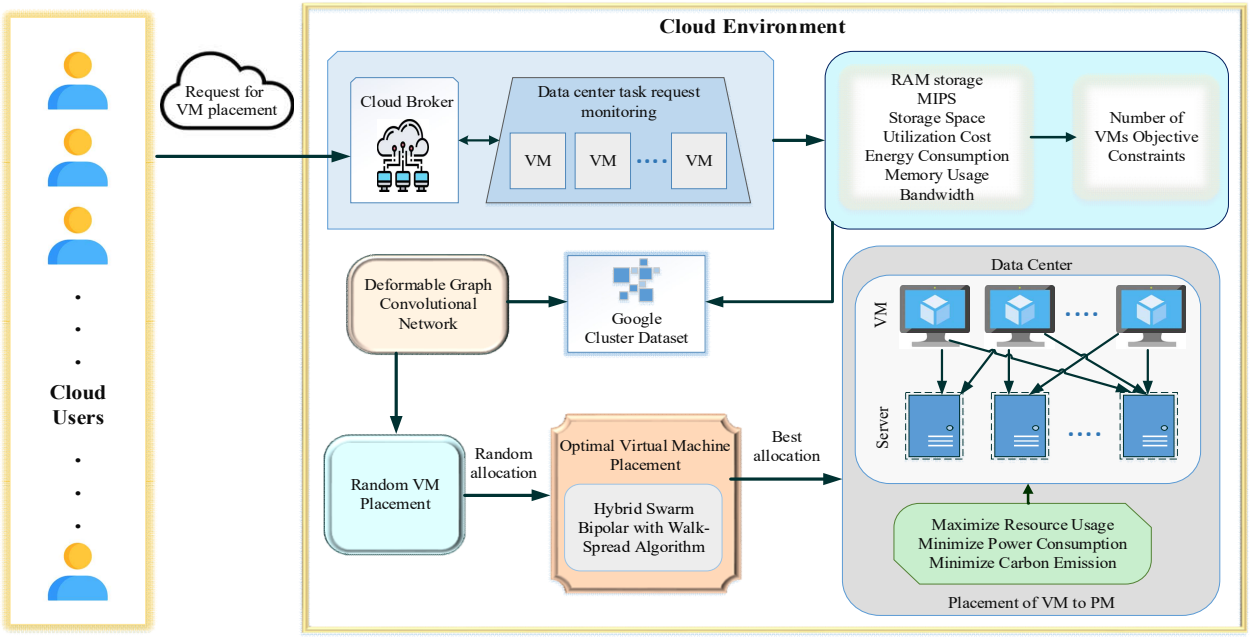

Fig. 1. Systematic Flow of the Proposed DGCN-SB-WSA Model.

3.1. System Model

The system model takes into account a data center architecture with physical machines (PMs) as well as VMs. $\{PM_1, PM_2, ..., PM_Q\} \in T$ *and* R $\{VM_1, VM_2, ..., VM_R\} \in W$ PMs and VMs have different setups. Since $\{v_1, v_2, ..., v_N\} \in V$, there are N users. For their applications, each user must configure their virtual machines on an economical in energy PM. The proposed design is a multi-objective VM placement method that maximizes resource utilization while minimizing power consumption and carbon emissions from data centers. One of three models is associated with each target. The Cloud Data Center (CDC) is home to a number of PMs that manage virtual machines. The Virtual Machine Manager (VMM) is responsible for scheduling and allocating VMs, and the optimization module uses a hybrid method to identify the optimal virtual machine location. The system paradigm for VM allocation is displayed in Fig. 2.

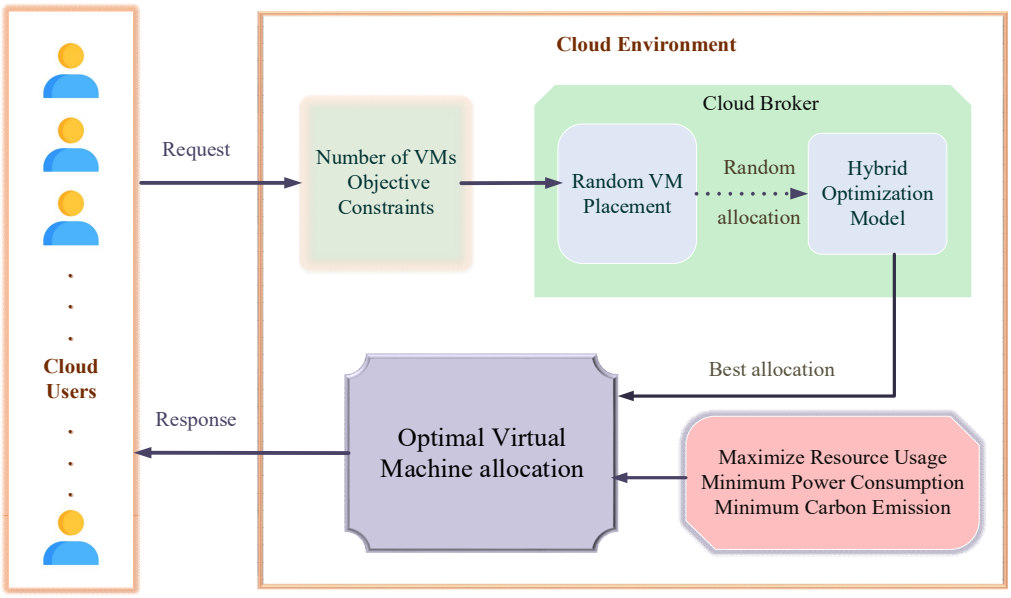

Fig. 2. System Model.

Resource Usage: Distribute CPU, RAM, and memory as efficiently as possible. Let PM_j^D, PM_j^N, PM_j^S represent the physical machine's RAM, memory as well as CPU capacities. Similarly, for j^{th} VMs, VM_j^D, VM_j^N, VM_j^S represent CPU, memory and RAM capacities. PM comes in two distinct modes: sleep mode and active mode. Here, $\eta = 1$ if PM is mode of operation, and $\eta = 0$ otherwise. Assume that w is a 2D matrix of size $Q \times R$ that shows the connection of VMs to PMs, where Q *and* R are the overall PM and VM counts. Set $x_{kj} = 1$ if PM_j servers on VM_j, and $x_{kj} = 0$ otherwise. A PM's specific resource consumption (memory, CPU, and RAM) is computed using equations (1), (2), and (3).

$$\mathrm{Res}Use_j^D = \frac{\sum_{k=1}^{R} x_{kj} \times VM_j^D}{PM_j^D} \tag{1}$$

$$\mathrm{Res}Use_j^N = \frac{\sum_{k=1}^{R} x_{kj} \times VM_j^N}{PM_j^N} \tag{2}$$

$$\mathrm{Res}Use_j^S = \frac{\sum_{k=1}^{R} x_{kj} \times VM_j^S}{PM_j^N} \tag{3}$$

where, $\mathrm{Res}Use_j^D$, $\mathrm{Res}Use_j^N$ *and* $\mathrm{Res}Use_j^S$ indicates the resource consumption in CPU, memory and RAM; Data center resource usage $\left(\mathrm{Res}Use_{DC}^S\right)$ is represented by Eq. (4), and overall resource use $\mathrm{Res}Use_{DC}$ is calculated using Eq. (5), where M is the entire quantity of services.

$$\mathrm{Res}Use_{DC}^S = \sum_{j=1}^{Q} \mathrm{Res}Use_j^S \qquad S \in \{N, S, D\} \tag{4}$$

$$\mathrm{Res}Use_{DC} = \int_{u1}^{u2} \frac{\mathrm{Res}Use_{DC}^D + \mathrm{Res}Use_{DC}^N + \mathrm{Res}Use_{DC}^S}{M \times \sum_{j=1}^{Q} \eta j} du \tag{5}$$

where, $\{N, S, D\}$ indicates the CPU, memory and RAM and $\sum_{j=1}^{Q} \eta j$ indicates the physical machines activation type.

Power Consumption: The CPU, RAM, storage, and the amount of operationally active networking devices all affect data center power usage. High operational costs are the result of these components' excessive use of energy. All of the PMs that primarily rely on integrated Dynamic Voltage Frequency Scaling (DVFS) energy-saving methods are considered in this paper. There are two possible states for the CPU: busy and sleep. The processing program and CPU

utilization rate determine how much energy is used in a congested scenario. CPU performs the least amount of work with the smallest clock cycle when in the sleep state. Not all of the CPU's components are in operating mode. Therefore, in equation (6) and (7), respectively, QX_j indicates the excessive use of energy of j^{th} VM and QX_{DC} indicates the entire data center power usage over a given period of time {t1, t2}, where the resource utilization of j^{th} PMs is indicated as $\mathrm{Re}sUse_j \in [0,1]$.

$$QX_j = \left(\left[QX_j^{\max} - QX_j^{\min}\right] \times QV_j + QX_j^{idle}\right) \tag{6}$$

$$QX_{DC} = \int_{u1}^{u2} \left(\sum_{j=1}^{Q} QX_j\right)\, du \tag{7}$$

where QX_j^{idle} indicates the idle power; $QX_j^{\max}$ is the maximum power; $QX_j^{\min}$ indicates the minimum power and QV_j is the utilization of server j, respectively.

Cooling Power: The cooling device component contributes to the data center's maximum power usage. Therefore, it cannot be disregarded because it limits service interruptions brought on by heat produced by servers. Examine how much electricity a cooling equipment uses in this research by examining traditional Computer Room Air Conditioning (CRAC) systems. The power consumption of the chiller unit is no longer impacted by the outside air temperature. A cooling device's performance is calculated using the Coefficient of Performance (CPow) measure in order to ascertain its ability to cool capacity. The ratio of (e/x) heat removed (for system force e) to the amount of effort required to remove the x heat is known as the CPow. A higher CPow number denotes more efficiency, meaning that a maximum amount of heat may be removed with the least amount of work. It is possible to represent the CPow metric as the following equation (8).

$$CPow = \left\{0.0068U_{TVQ}^2 + 0.0008U_{TVQ} + 0.458\right\} \tag{8}$$

where, U_{TVQ} indicates the variation in temperature; then overall effect on the environment $UDG(u)$ formulated at time u as equation (9).

$$UDG(u) = QVF_e \times DGS \times QT \tag{9}$$

where, QVF represents the effectiveness of dynamic power utilization and the data center's QVF is calculated as $QVF_e = (Amontof\ electicalenergy/IT\ power)$, amount of electrical energy of the facility DGS is $QT + PQ$, here, PQ indicates the total overhead power is $PQ = QT/CPow$ and QT shows how much electricity each PM in the data center uses overall.

Carbon Emissions: The reduction of power use and cooling needs by strategically placing virtual machines is expressed in equation (10).

$$CE = \min \sum_{k=1}^{n} F_k . y_{jk} \tag{10}$$

where, CE signifies the carbon emission and F_k indicates the PM k^{th} carbon emission factor, respectively.

Trade-off between several objectives: By activating the smallest amount of active servers, optimizing resource utilization and lowering energy use are likely to optimize the VM consolidation process, even if the quantity of active PMs in the data center impacts the amount of carbon emissions. High power consumption and thermal power dissipation are the primary causes of data centers' overall operating costs. Based on their resource processing capacities, resource utilization must be optimized to reduce the power consumption of both idle and functional PMs. The workload has complete impact on power usage. Static power loss can be experienced by idle PM. Therefore, the burden must be divided among fewer active PMs in order to achieve the least power needs.

3.1.1. Problem Formulation

The crucial sets of requirements that must be met during VMP are listed below; they have been developed using the provided equations (11).

$$
\begin{aligned}
D_1 &= \sum_{k=1}^{R} VM_k^D \times x_{jk} \le PM_j^D \qquad \forall j \in 1,2,\ldots,P \\
D_2 &= \sum_{k=1}^{R} VM_k^S \times x_{jk} \le PM_j^S \qquad \forall j \in 1,2,\ldots,P \\
D_3 &= \sum_{k=1}^{R} VM_k^N \times x_{jk} \le PM_j^N \qquad \forall j \in 1,2,\ldots,P
\end{aligned} \tag{11}
$$

where, x_{jk} is the j^{th} VM's mapping on the j^{th} physical machine, and VM_j^D, VM_j^N, VM_j^S are the j^{th} VM's secondary storage requirements, RAM and CPU capacity (in MIPS). Likewise, PM_j^D, PM_j^N, PM_j^S represent the physical machine's needs secondary storage requirements. The proposed structure is a VMP with many objectives. It offers the best virtual machine allocations, ensuring quick user task completion and effective physical resource consumption with the least possible carbon footprint. Reduce power consumption and carbon emissions to maximize resource use. Equations (12) are used in the problem formulation.

$$\upsilon = \sum_{j=k=1}^{Q,R} x_{jk} \tag{12}$$

where, $Minimize\ g\left(QX_{DC}(\upsilon)\right), g\left(CE_{DC}(\upsilon)\right)$ and $Maximize\ g\left(\mathrm{Re}\,sUse_{DC}(\upsilon)\right)$; υ indicates the distribution of R VMs among Q PMs in accordance with goals like maximizing SV and minimizing QX and DF, respectively.

3.2. Data Preparation

Google provided the Google Cluster Dataset (GCD) [27], a publicly accessible dataset that includes workload traces from Google's cloud data centers. Workload scheduling, resource administration and online computing research all make extensive use of it. Researchers may examine and create effective VM deployment, load balancing and power optimization strategies thanks to the dataset's insights into actual cluster resource usage. A unique data preparation procedure is thus carried out in advance, and every virtual machine is outfitted with a multi-resource predictor of its own. The two stages of data preparation are attribute extraction and aggregation, which are followed by normalization. In order to anticipate the resource allocation during the next scheduling period, useful properties (connected to resources, such as CPU and memory utilization of a task at particular VM) are extracted from both historical and live data and aggregated per unit time. To ensure consistency in resource distribution, data is normalized using min-max normalization equation (13), which standardizes the sum of the values of each resource and attribute.

$$\hat{E} = \frac{E_j - E_{\min}}{E_{\max} - E_{\min}} \tag{13}$$

where, $\hat{E}$ denotes the normalized vector; $E_{\min}$ and $E_{\max}$ denote the input data set minimum and maximum values, respectively. The normalized vector is a set that includes every normalized input data that reflects certain resource utilization. Multiple input vectors are fed into the Deformable Graph Convolutional Network's input layer, depending on the amount of resources taken into consideration, after a distinct normalized vector for each resource is calculated.

3.3. Deformable Graph Convolutional Network-Based Virtual Machine Deployment Resource Allocation in Cloud Environment

This section involves the effective implementation of a convolutional network for the Deformable Graph Convolutional Network (DGCN) [24] to manage resources and position virtual machines in cloud services. The primary approach is to visualize the cloud architecture as a graph, with VMs and PMs as nodes and the interconnections between them (such as resource dependencies and network connections) as edges. By taking into account both short- and long-range dependencies in the cloud environment, Deformable GCNs allow us to make adaptive placement decisions. The procedures utilizing the five essential equations from the original Deformable GCN framework are as follows.

3.3.1. Representing the Cloud Infrastructure as a Graph

A PM or VM is represented by each node in the graph and their interactions are defined by the edges. Nodes' characteristics include: Available CPU, memory, storage, bandwidth and historical workload data are all important for PMs; and requested CPU, memory, storage, and network demand for virtual machines. The current graph convolution operation is applied to node w in equation (14).

$$g_w^{(m)} = \varpi\left(X^{(m)}.\ AGGREGATE\ \left(g_v^{(m-1)}: v \in M\, f(w)\right)\right) \tag{14}$$

where, $g_w^{(m)}$ indicates the node $w's$ hidden representation of feature vector at layer m; ϖ represents the non-linear activation function; $X^{(m)}$ signifies the m^{th} layer learnable weight matrix; $AGGREGATE\ (\cdot)$ denotes the aggregation function that collects data from $w's$ neighbors such as mean, sum, max or attention-based; $w's$ indicates the neighbour $v's$ feature vector from the preceding layer $m-1$ and $v \in M\, f(w)$ here, $M\, f(w)$ denotes the collection of adjacent nodes v linked to node w. By combining data from nearby nodes (PMs and VMs), this equation simulates resource dependencies. This makes it possible to depict the dynamic resource needs of the cloud infrastructure more accurately, which serves as the basis for building the latent neighbourhood graph.

3.3.2. Constructing the Latent Neighbourhood Graph

To efficiently describe the relationships between different VMs and PMs, it computes the k-nearest neighbors (kNN) network based on resource similarity. It creates a latent graph by connecting nodes with comparable usage patterns instead of using pre-established physical connections. It captures patterns in resource distribution in cloud environments by smoothing input features over multiple iterations. It is calculated in equation (15).

$$f_w^{(m)} = \frac{1}{d\tilde{e}g(w)} \sum_{v \in \tilde{M}(w)} f_v^{(m-1)} \tag{15}$$

where, $f_w^{(m)}$ denotes the update of the edge feature for node w at layer m; $f_v^{(m-1)}$ indicates the edge feature of adjacent node v from the preceding layer $m-1$; $M(w)$ signifies the collection of nearby nodes v linked to w; $\sum_{v \in \tilde{M}(w)} f_v^{(m-1)}$ denotes the summation of edge characteristics from all neighbors of node w in the previous layer; $d\tilde{e}g(w)$ indicates the degree function of node w, which denotes the number of adjacent nodes and $\frac{1}{d\tilde{e}g(w)}$ indicates the normalization term that guarantees the average of the aggregated edge characteristics. In order to measure VM-PM interactions, then build relation vectors, which capture resource affinities and dynamic dependencies between virtual and physical machines.

3.3.3. Computing Relation Vectors for VM-PM Interactions

A relation vector is calculated based on the node embeddings of VMs and PMs in order to describe their dynamic interactions. This aids in determining if a VM and a PM are a good fit for placement. In order to provide better placement choices, equation (16) illustrates how PMs and VMs are comparable in terms of their resource availability and needs.

$$s_{v,w} = \begin{cases} \left[\frac{\gamma_v - \gamma_w}{\|\gamma_v - \gamma_w\|^2} \middle\| 0\right] \in \Re^{d_{\gamma+1}}, & if\ \gamma_v \neq \gamma_w \\ [0,0,\ldots,1] \in \Re^{d_{\gamma+1}}, & otherwise \end{cases} \tag{16}$$

where, $s_{v,w}$ indicates the connection or transformation function between the graph's nodes v and w; $\|\gamma_v - \gamma_w\|^2$ denotes the measure of the difference between the feature representations of nodes $v\ and\ w$ is the squared Euclidean distance; $\gamma_v - \gamma_w$ denotes the direct difference between nodes $v's$ and $w's$ feature vectors, which represents their relative displacement in the feature space; $[0,0,...,1]$ is when $\gamma_v = \gamma_w$, a present vector likely an indicator vector is employed to make sure that no transformation takes place and $\gamma_v \neq \gamma_w$ denotes the making certain that no transformation takes place in these situations. Then, using adaptive message passing using deformable graph convolution, and improve node embeddings by dynamically modifying the graph topology in response to VM-PM interactions.

3.3.4. Adaptive Message Passing Using Deformable Graph Convolution

The message transfer between PMs and VMs is adaptively adjusted by DGCNs according to their relationships. They pick up adaptable patterns to record significant interactions, in contrast to set aggregation procedures. The Cloud

Data Center's (CDC) system efficiency and resource allocation are enhanced by this dynamic metamorphosis. It is calculated in equation (17).

$$z_w = \sum_{v \in \tilde{M}(w)} h_{deform}\left(s_{v,w} . \Delta_l(f_w) g_v\right) \tag{17}$$

where, z_w following the application of the deformable graph convolution, the output feature representation of node w; $\sum_{v \in \tilde{M}(w)}$ denotes the node $w's$ dynamically sampled neighbourhood; $h_{deform}\left(s_{v,w} . \Delta_l\left(f_w\right)\right)$ denotes the kernel function that ascertains how neighbour v affects node w; g_v signifies the node w is updated by aggregating the adjacent node $v's$ feature representation; $h_{deform}\left(s_{v,w}\right)$ specifies a learnable transformation function that adjusts the message exchange according to the degree to which a PM's capacity and a VM's requirements are comparable. The model is better able to represent intricate connections between VMs and PMs because of this adaptive approach, which guarantees that the message passing mechanism stays dynamic and context-aware.

3.3.5. Aggregating Multiple Graph Representations for Final Placement Decision

The management of VMs needs to be flexible as cloud workloads are continuously fluctuating. DGCNs aggregate numerous neighbourhood graphs to capture the dynamic interactions between PMs and VMs. This adaptive approach allows the system to adjust to workload changes effectively. DGCNs learn and enhance the linkages continuously to improve the decision-making process for VM placement. As a result, they enhance the responsiveness and flexibility of the cloud infrastructure. The final VM placement is given in equation (18).

$$\tilde{g}_w = \sum_{m=0}^{M+1} t_w^{(m)} \tilde{z}_w^{(m)} \tag{18}$$

where, $\tilde{g}_w$ indicates the final representation; $\tilde{z}_w^{(m)}$ denotes the node representation in layer m; $t_w^{(m)}$ denotes the attention score for layer l, and m M signifies the learnable attention parameter. This equation balances data from several neighborhood graphs to ensure that both short-term and long-term resource requirements will be considered to make a placement choice. DGCNs capture both local and global resource restrictions and allow adaptive VM placement by dynamically modeling VM-PM connections. This approach can balance workload distribution and enhance the usage of cloud environments by ensuring effective resource allocation. After VM allocation, a hybrid optimization approach is used to optimize VM placement in the DGCN model.

3.4. Hybrid Swarm Bipolar with Walk-Spread Algorithm for Optimal VM Allocation

The Hybrid Swarm Bipolar with Walk-Spread Algorithm: This approach integrates a Bipolar Swarm Algorithm [25] with the WSA [26] following the DGCN phase to enhance the allocation of VMs. BSA uses the attraction and repulsion of dual forces while performing effective searches. The WSA makes use of the local intensification-spread and global exploitation-walk as the ideas for enhancing search.

Rationale for Selecting SB-WSA: The reasons for choosing the SB-WSA over other swarm-based and metaheuristic algorithms, such as PSO, ACO, and GA, are that it achieves a better balance between exploration and exploitation to effectively avoid premature convergence. Its bipolar decision mechanism promotes diversified candidate solutions, while the walk-spread strategy allows for adaptive refinement of the VM allocations in dynamic cloud environments. This makes SB-WSA more resilient when faced with fluctuating workloads and more capable of optimizing multiple conflicting objectives-resource utilization, power efficiency, and carbon emission reduction-compared to traditional optimizers. Thus, SB-WSA provides a robust and energy-efficient search mechanism for refining the DGCN-predicted VM placements.

Step 1: Initialization (Generating Initial VM Allocation Candidates)

In the search space, every VM placement candidate $v_{j,k}$ is evenly initialized, as expressed in equation (19).

$$v_{j,k} = c_{m,k} + s(0,1) \ . \ c_{x,k} \tag{19}$$

where, $c_{m,k}$ *and* $c_{x,k}$ indicates the random uniform value that ensures variation in VM placement and the search space's border range. Equation (20) is used to calculate the range width.

$$c_{x,k} = c_{v,k} - c_{m,k} \tag{20}$$

where, $c_{x,k}$ indicates range width. Equation (21) provides the global best solution only when VM allocations are improved, as guaranteed by a stringent acceptance requirement.

$$v_c' = \begin{cases} v_j, & if\ g(v_j) < g(v_c) \\ v_c, & otherwise \end{cases} \tag{21}$$

where, $v_c^{'}$ indicates the global best solution and $g(v)$ denotes the objective function measuring VM allocation efficiency.

Step 2: Swarm Bipolar Algorithm for VM Allocation

SBA in contrast to conventional swarm intelligence introduces two opposing forces. Attractive force advances solutions $t_{j,k}$ in the direction of the most well-known solution. By pushing best solutions away from bad ones, the repulsive force prevents $d_{j,k}$ premature convergence. These forces are dynamically followed by each VM allocation in Eq. (22), (23) and (24).

$$t_{j,k} = t_{c,k} + s_1(v_{c,k} - v_{j,k}) \tag{22}$$

$$d_{j,k} = t_{d,k} + s_1(v_{j,k} - v_{worst,k}) \tag{23}$$

$$v'_{j,k} = v_{j,k} + s(0,1)\ .\ (t_{j,k} - d_{j,k}) \tag{24}$$

where, $t_{c,k}$ and $t_{d,k}$ denotes the weights balancing attraction and repulsion; $v_{c,k}$ signifies the global best VM placement; $v_{worst,k}$ is the worst VM placement (to be avoided) and $v'_{j,k}$ indicates the best solution. This equation guarantees that while poor VM placements are removed, good placements are strengthened.

Step 3: Walk Phase for Refining VM Allocations

The first walk, Towards the Ideal VM Position, by adhering to the best placement trend, this procedure guarantees convergence. Only if the revised virtual machine allocation results in better performance will it be approved. Every VM location advances toward the global best solution found so far, as given in Eq. (25).

$$v'_{j,k} = \begin{cases} d_k, & if\ g(d_k) < g(v_{j,k}) \\ v_{j,k}, & otherwise \end{cases} \tag{25}$$

where, d_k indicates the Candidate solution (new possible VM allocation); $g(d_k)$ and $g(v_{j,k})$ Objective functions, respectively. Second Approach, Diversification of VM Placement a secondary approach is introduced by choosing two alternate locations to maintain diversity. Then, VM placements move toward the target in Eq. (26).

$$d_k = v_{j,k} + s(0,1).(v_{u,k} - 2v_{j,k}) \tag{26}$$

where, $v_{u,k}$ indicates the alternative location. This walk prevents stagnation and maintains exploration capabilities.

Step 4: Spread Phase (Neighbourhood Optimization of VM Placement)

The spread phase further optimizes the VM allocations using two levels of neighbourhood searches. First spread wider search for exploration in Eq. (27).

$$d_{t,l,k} = v_{j,k} + s(-1,1).c_{x,k}.\left(1 - \frac{u}{u_n}\right) \tag{27}$$

where, $d_{t,l,k}$ denotes the Spread Phase and $\left(1-\frac{u}{u_n}\right)$ indicates the gradual reduction in the search radius over iterations, the spread-generated children's best contender for VM allocation.

Step 5: Exploitation Phase (Narrower Search)

Through the exploration of minute changes in VM location, a more focused search within a narrower neighbourhood radius improves precision. With little variations, this fine-tuning guarantees the best possible resource allocation. It is given in Eq. (28).

$$d_{t,l,k} = v_{j,k} + 0.01.s(-1,1).c_{x,k}.\left(1-\frac{u}{u_n}\right) \tag{28}$$

where, $c_{x,k}$ indicates the total iteration count; $0.01.s(-1,1)$ signifies the VM placements are fine-tuned locally, with the search radius being just 1% of the initial spread. A strict acceptance criterion only approves novel solutions that enhance the objective function.

The flowchart of SB-WSA is shown in Fig. 3. The detailed process for obtaining optimal solutions:

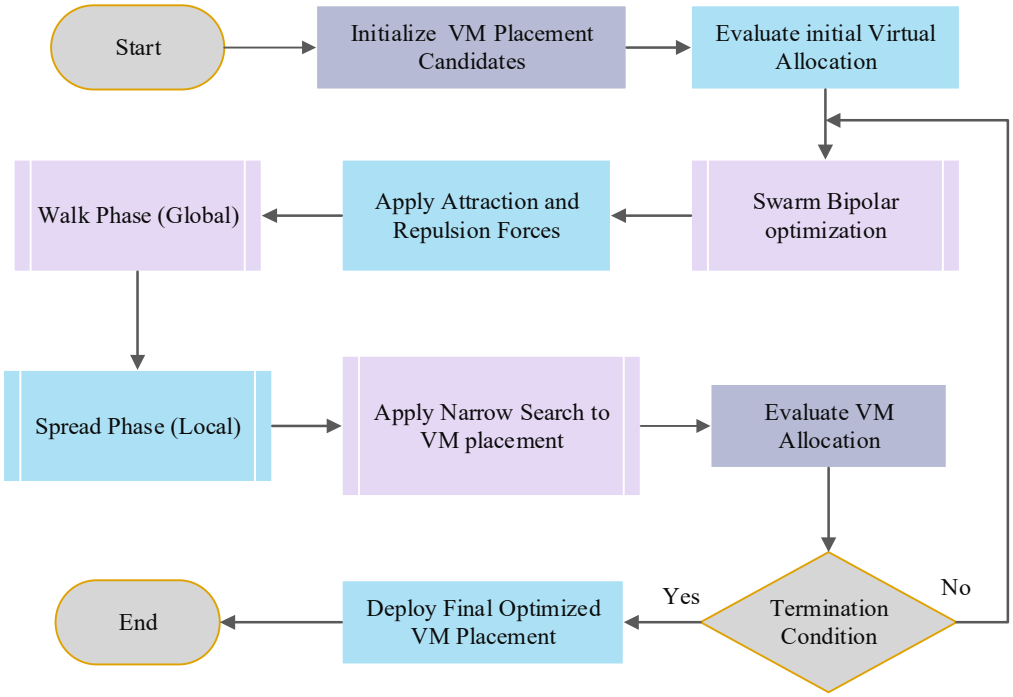

Fig. 3. Flow Diagram of the Hybrid Swarm Bipolar Walk-Spread Algorithm.

Step 6: Termination

The walk and spread techniques are iterated through by the algorithm until convergence is reached. When there are no more notable advancements in VM placement, convergence is declared. Through constant solution improvement, the procedure guarantees optimal allocation.

Algorithm 1. Hybrid Swarm Bipolar Walk-Spread Algorithm (SB-WSA)

Input: Set of Virtual Machines (VMs), Physical Machines (PMs), Resource constraints
Output: Optimal VM-to-PM allocation minimizing power and carbon emission while maximizing resource utilization
1. Initialize population of candidate VM allocations randomly.
2. Evaluate objective function for each candidate based on energy, utilization, and carbon emission.
3. *while (termination condition not met)* do
a. Apply Swarm Bipolar Algorithm (SBA)
– Compute attraction toward best solution.
– Compute repulsion from worst solution.
– Update positions of VM allocations.
b. Apply Walk Phase

– Move each candidate toward global best.
– Diversify using alternative placement candidates.

c. Apply Spread Phase
– Conduct broad search for unexplored allocations.
– Gradually narrow the search radius.

d. Exploitation Phase
– Fine-tune allocations through small local adjustments.

e. Update best solution.

4. *end while*

5. Return the global best VM-to-PM allocation.

Note: The pseudocode summarizes the mathematical process defined in Eqs. (19–28), presenting the SB-WSA optimization steps in a clear, reproducible form.

3.5. Operational Design and Time Complexity Analysis

The SB-WSA mechanism's overall operational summary is interpreted by the proposed algorithm. Six distinct phases can be used to describe the algorithm's functionality. It can use a random allocation approach to do VM-to-PM mapping in step 1. The VM allocations are refined in step 2 using SBA, in step 3 using WSA, and in steps 4 and 5 in the Spread Phase. Step 5 involves using SB-WSA to identify an optimal set of virtual machine allocations that meet all of the aforementioned goals. Here, $Q, R, c_{x,k}$ *and* u are initialized in step 1. Random virtual machine allocations are obtained in steps 2-5, where time complexity depends on Q *and* R values equal to $\theta(Q \times R)$. Each placement must adhere to the resource limits outlined in Eq (11). Next, use SBA with time complexity $\theta(Q \times R)$ to allocate VMs that weren't assigned during the first deployment phase. Three objectives' values are calculated using Equations (5), (7) and (9). To obtain the best allocations, the optimization process is carried out in step five. Equation (19) is used to initialize the number of swarm bipolar with walk-spread equal to the random communities generated by the first phase of this technique. With time complexity $\theta(pc)$ in (20), where pc is the total number of objectives, the Walk Phase yields the best solution $d_{t,l,k}$ when employing WSA, the neighbourhood optimization of VM Placement value is calculated in Equation (26). Both local and global searches are conducted based on the value. The procedure can be repeated until the termination requirements are met. The entire optimization procedure will be time-consuming $\theta(c_{x,k})$ and difficult. Therefore, the total time complexity for the best VMP is $\theta((pc) \times Q \times R \times c_{x,k})$.

4. Empirical Results and Discussion

The research and numerical findings of the proposed hybrid Swarm Bipolar with Walk-Spread Algorithm for ideal virtual machine allocation are shown in this section. The performance is compared with existing metaheuristic techniques and assessed using a variety of benchmark functions. The outcomes show how well the algorithm works to increase resource efficiency and lower energy usage in cloud systems.

4.1. Experimental Set-up

The analysis and findings of the DGCN-SB-WSA approach are covered in this section. Two Intel® Xeon® Silver 4114 CPUs (10 cores/20 threads each), are part of the server system utilized for experimental assessments. The computing device runs 64-bit Ubuntu 16.04 LTS and features 128 GB of main RAM. The Python 3.7 environment and the necessary libraries for data processing, modelling and analysis are used to run the simulations. Table 2 shows the simulation setup.

Table 2. Simulation Parameters with Ranges.

Parameter	Ranges
Quantity of Virtual Machines	200-1200
Quantity of Physical Machines	60-700
Bandwidth	1500 - 3500 B/S
CPU MIPS	2500
Number of PEs per machine	1
Dimension	50, 75
Memory Size	2048-12576 MB
Consumption of Processor Power	130W – 240W
Consumption of Cooling Devices Power	400W – 900W
Power Consumption by RAM	10W – 30W
Power Consumption by Storage	35W – 110W
Power Consumption by Network	70W – 180W
Extra Components Power Usage	4W – 34W

The relatively high efficiency (e.g., 98% resource utilization and 27 kW power consumption) results from a controlled experimental configuration using the Google Cluster Dataset. During testing, workloads were normalized using min–max scaling and synthetic load balancing was applied to maintain uniform task distribution across PMs. Each experimental run was averaged over 10 trials to minimize stochastic bias. The physical machine configurations were selected for high-density energy-efficient operation, explaining the lower power readings compared to typical heterogeneous environments. While these settings demonstrate the upper performance bound of the proposed model, future work will include validation on diverse datasets such as PlanetLab and real hybrid-cloud traces to confirm generalizability.

4.2. Evaluation of VM placement method with Performance metrics

This paper assesses the efficacy of the proposed DGCN-SB-WSA technique and contrasts it with alternative methods by utilizing critical performance criteria such as resource usage, carbon emission, power consumption, number of active servers, carbon footprint and response time. The Google Cluster Dataset (GCD) results are analysed and compared with conventional methods to demonstrate the proposed approach's improvements in VM allocation.

4.2.1. Resource utilization (RU)

Resource usage is the ratio of a system's active use of its computational resources (CPU, memory and storage) to its total capacity. It is expressed in Eq. (29).

$$\text{Re}source\ usage = \frac{Used\ resource}{Total\ available\ resources} \times 100 \tag{29}$$

4.2.2. Power consumption (PC)

PC is the total amount of energy used by servers that are actively doing tasks. Reduced power usage is a factor in energy efficiency. It is given in Eq. (30).

$$Power\ consumption = \sum_{j=1}^{m} Q_{idle} + (Q_{\max} - Q_{idle}) \times V_j \tag{30}$$

Where, Q_{idle} indicates the idle power; $Q_{\max}$ is the maximum power; V_j is the utilization of server j, respectively.

4.2.3. Carbon emission (CE)

Carbon emissions are the quantity of CO_2 that data centers release as a result of their energy use. Power use and the energy source's carbon intensity have an impact. It is calculated in Eq. (31).

$$Caebon\ Emission = Power\ consumed - Emission\ factor \tag{31}$$

4.2.4. Number of Active Servers (NAS)

The total number of servers that are now engaged in tasks, energy usage can be decreased by optimizing running servers, is expressed in Eq. (32)

$$N_{active} = \sum_{j=1}^{m} T_j \tag{32}$$

Where, T_j indicates the 1 if the server is active and 0 otherwise, respectively.

4.2.5. Carbon footprint (CF)

The cumulative amount of greenhouse gas emissions brought on by computer infrastructure. Emissions of CO_2 equivalent are used to measure it. It is given in Eq. (33).

$$Carbon\ \ Footpr\text{int} = \sum_{t=1}^{T} Carbon\ Emisssion_t \tag{33}$$

4.2.6. Response time (RT)

It is the amount of time (including network latency, execution time, and queue time) needed to process a request and return the result. It is given in Eq. (34).

$$\text{Re}sponse\ time = T_{completion} - T_{arrival} \tag{34}$$

where, $T_{completion}$ indicates the time a task finishes execution and $T_{arrival}$ signifies the time taken for VM allocation, respectively.

4.3. Evaluation of the proposed model's performance

The proposed DGCN-SB-WSA is tested utilizing important performance indicators as resource usage, carbon emission, power consumption, NAS, carbon footprint and response time. To evaluate the model's effectiveness in VM placement, it is put through a series of workload scenarios. Throughout several iterations, convergence behavior and solution stability are noted. The assessment centers on how well the search tactics optimize the distribution of resources. Four distinct VM categories and three distinct PM types are taken into account in this research. Table 3 provides illustrations of the PM-VM combinations.

To evaluate the robustness and generalizability of the proposed DGCN–SB-WSA model, supplementary experiments were also conducted using the PlanetLab dataset and synthetic workload traces generated with variable resource-intensity profiles. The PlanetLab dataset captures the geographical distribution of nodes and high volatility in workloads, while the synthetic data simulate controlled variations in CPU, memory, and bandwidth utilization. The results obtained from these extra datasets show performance trends similar to those obtained with the Google Cluster Dataset (GCD), confirming that the model proposed is able to preserve efficiency, scalability, and adaptability for diverse and dynamic workload conditions.

Table 3. PM and VM Configuration.

Type	PM-VM	PE	MIPS	RAM (GB)	Storage (GB)	Q_{max}	Q_{min}/Q_{idle}
PM	PM_1	2	2660	4	160	135	93.7
PM	PM_2	4	3067	8	250	113	42.3
PM	PM_3	12	3067	16	500	222	58.4
VM	VM_S	1	1000	0.5	70	-	-
VM	VM_M	2	1500	3	90	-	-
VM	VM_L	3	2000	4	100	-	-
VM	VM_XL	4	2500	4	150	-	-

Table 3 depicts the configuration of the VMs and PMs. In total, each PM includes a number of PEs, MIPS, RAM, storage, and power consumption numbers Qmax, Qmin/Qidle. Among the PMs, which are all different from one another in computational powers, PM3 is the most powerful one. The virtual machines do not have any power consumption numbers, but they do have variable resources to support different workloads.

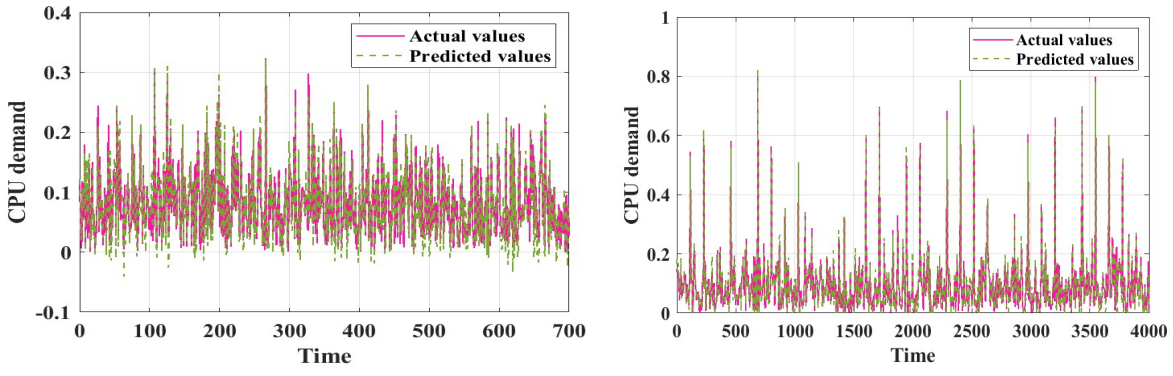

Fig. 4. GCD–CPU Performance on (a) SB-WSA (PWS = 15 min) and (b) (PWS = 75 min).

The changes in CPU demand due to time are reflected by comparing actual and expected figures in Fig. 4(a). The graph shows the fluctuating computational demand, which establishes the efficacy of the proposed SB-WSA model in tracking and forecasting resource usage. Fig. 4(b) shows CPU demand for a longer period to highlight the difference between actual and expected numbers. The fact that expected and actual numbers agree establishes the correctness of the proposed method in calculating the computing demand to facilitate the optimal deployment of virtual machines. In this process, the time span used for estimating the trend in CPU usage is 75 minutes, which is the PWS.

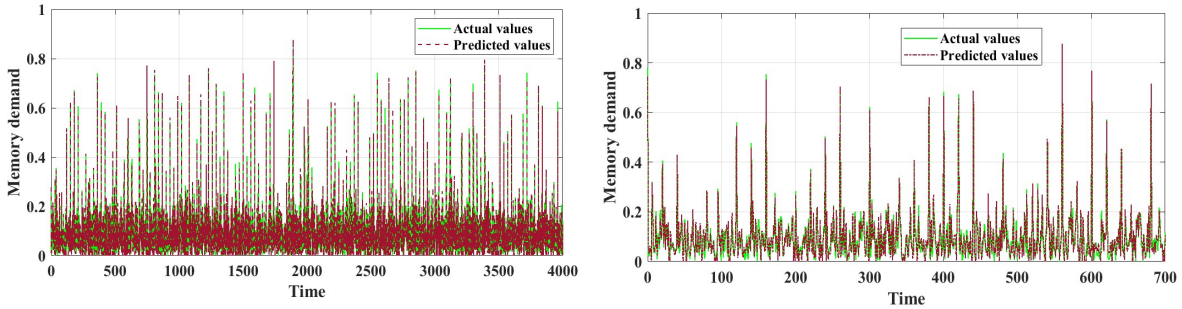

Fig. 5. GCD–Memory Performance on (a) SB-WSA (PWS = 15 min) and (b) (PWS = 75 min).

Fig. 5(a) plots real and expected values to illustrate how memory demand varies over time. Memory demand levels are plotted versus time intervals on the y and x axes, respectively. Dynamic changes in the utilization of memory as workload patterns change can be seen from this graph. The actual and expected values of the SB-WSA model have aligned, hence indicating the accuracy of the model in terms of memory demand prediction. Fig. 5(b) contrasts actual and expected values to show how memory demand varies over time. It illustrates the accuracy of the SB-WSA model in predicting resource utilization by highlighting the differences in the pattern of memory usage. The differences between real and anticipated values are not that high, so this means the model is accurate in predicting memory.

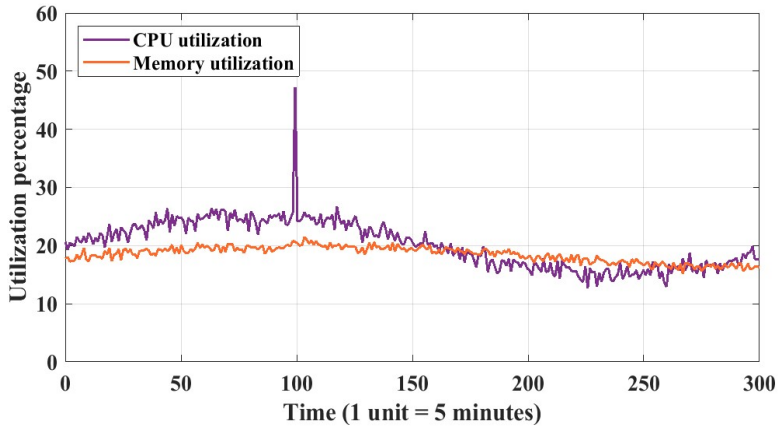

Fig. 6. Resource Usage of a Randomly Selected VM from GCD VM Traces.

Figure 6 shows the CPU and memory usage patterns of a randomly chosen virtual machine from Google cluster traces. Time is shown on the x-axis in five-minute increments, while the usage percentage is shown on the y-axis. With sporadic spikes and swings, this illustrates changes in CPU and memory usage across the observed time period. This analysis helps in understanding workload behavior and resource demand patterns for better virtual machine placement and management in cloud settings.

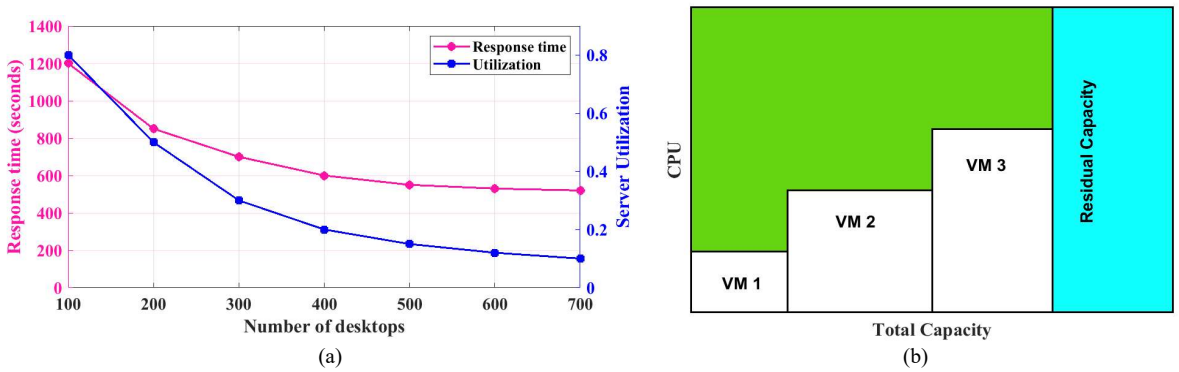

Fig. 7. (a) Response Time and (b) Resource Wastage.

Two important performance measures, response time, and server utilization, are plotted against the number of VMs in Fig. 7(a). As the number of computers increases, the response time falls off dramatically, as does the server

utilization. It denotes that, with a higher spreading of workloads, the system becomes more efficient by allowing fewer delays in processing. The residual capacity utilization and resource distribution across different VMs are depicted in Fig. 7(b). Distinct sections are drawn for the allocation to VM 1, VM 2, and VM 3, while the Total Capacity depicts the overall capacity. The unallocated portion shows the remaining capacity after virtual machine allocation. This visualization is intended to point out the problem of resource wastage and emphasize the importance of an efficient VM placement strategy.

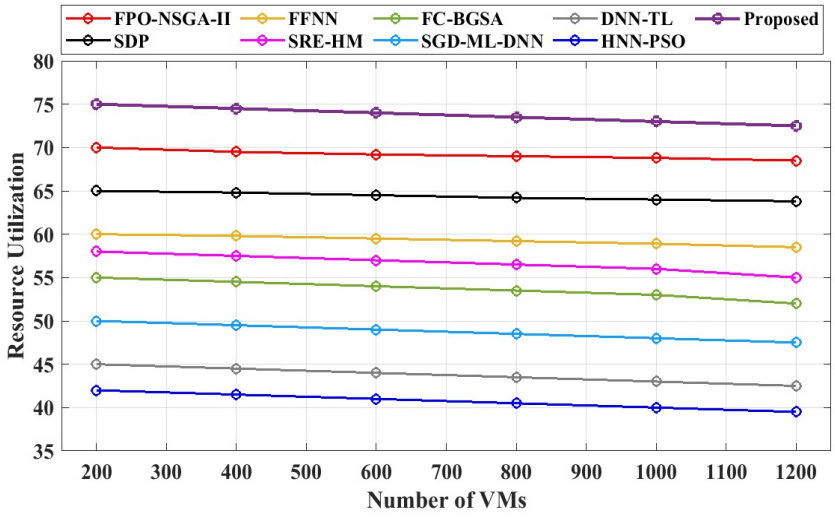

Fig. 8. Performance Analysis of Resource Utilization.

Among them, the proposed DGCN–SB-WSA method consistently achieves higher and more stable utilization compared to baseline approaches such as FPO-NSGA-II, SGD-ML-DNN, SDP, and HNN-PSO, which further shows that it has better scalability and efficient resource management while facing dynamic workloads.

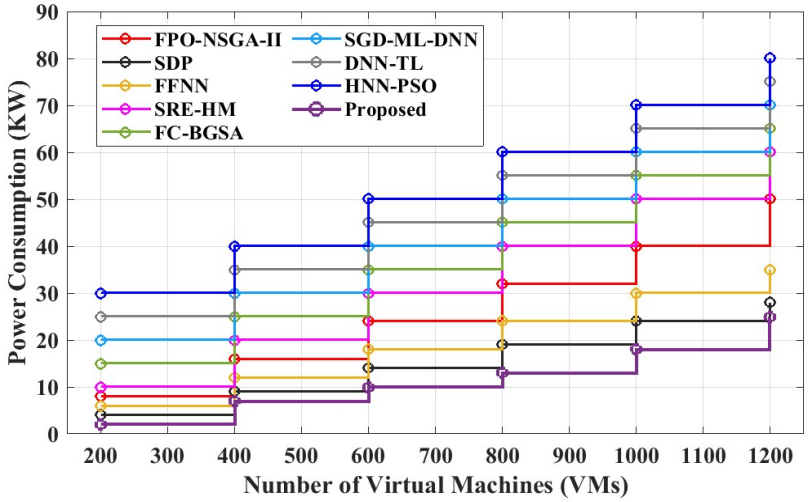

Fig. 9. Analysis of Power Consumption vs. Number of VMs.

In this regard, the results depict that the proposed DGCN–SB-WSA model has lower power consumption values than the competing algorithms: FPO-NSGA-II, SGD-ML-DNN, SDP, FFNN, SRE-HM, DNN-TL, HNN-PSO, and FC-BGSA, at all load levels, showing the superior energy efficiency of the proposed model.

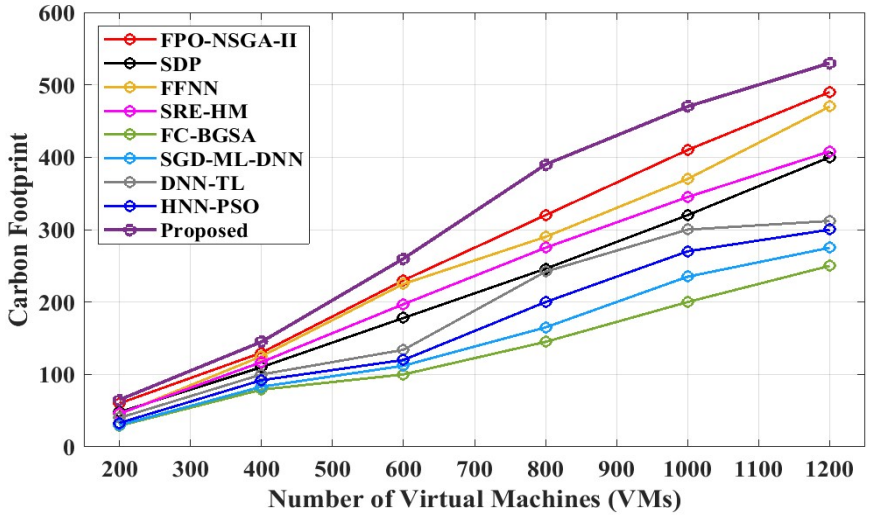

Fig. 10. Analysis of Carbon Footprint.

Although the proposed DGCN–SB-WSA model achieves higher resource utilization and task handling capability, its carbon emission is slightly higher due to higher energy throughput, which justifies the trade-off between performance and environment sustainability.

4.4. Comparative Analysis

Various important performance metrics-resource usage, carbon emission, power usage, NAS, carbon footprint, and response time-are considered to evaluate the performance of the proposed DGCN-SB-WSA architecture for VM placement. FPO-NSGA-II [16], SDP [17], FFNN [18], SRE-HM [19], FC-BGSA [20], SGD-ML-DNN [21], DNN-TL [22] and HNN-PSO [23] are compared to the proposed model on the GCD.

Table 4. Performance Comparison on GCD.

Methods	RU (%)	PC (kW)	CE (kg CO_2)	NAS	CF (kg)	RT (s)
FPO-NSGA-II [16]	72	55	300	280	450	90.0
SDP [17]	68	50	280	270	420	85.0
FFNN [18]	66	48	260	260	400	50.90
SRE-HM [19]	64	46	250	255	390	78.0
FC-BGSA [20]	62	44	240	250	380	75.0
SGD-ML-DNN [21]	60	42	230	245	370	7.20
DNN-TL [22]	58	40	220	240	360	70.0
HNN-PSO [23]	55	38	210	235	350	68.0
DGCN-SB-WSA (Proposed)	98	27	180	175	300	23.1

Table 4 presents the efficiencies of different methods for cloud virtual machine deployment in GCD. The DGCN-SB-WSA model turns out to be much better compared to other existing methods, as it can reach the maximum resource utilization (98%), with a high carbon emission reduction (180 kg CO_2) and power consumption (27 kW), while adopting the least active servers (175) for a low carbon footprint (300 kg). Most importantly, compared with other methods, the proposed model has the lowest response time of 23.1 s, indicating high efficiency in task execution.

4.5. Statistical Analysis

This dataset is generated statistically by the proposed algorithm, which is based on an assisted randomization technique. The algorithm's relevance is quantitatively evaluated by ANOVA and the Duncan's Multiple Range test.

4.5.1. ANOVA Test

Instead of finding out which is the best performing algorithm, the ANOVA test shows whether the performance differences are significant. Although it cannot determine which algorithms perform better, the ANOVA test enables us to be certain that there is a significant difference between the algorithms. The ANOVA statistical analysis is shown in Table 5.

Table 5. Analysis of 200 VM Instances Using ANOVA for ONVG.

Source Factor		Sum of Squares	Degrees of Freedom	Mean Square	F-value	Sig (value of p)
Spacing (SP)	Between Groups	0.560	6	0.193	28.427	0.000
	Within Groups	0.094	23	0.006	-	-
	Overall	0.613	27	-	-	-
ONVG	Between Groups	563.498	6	147.822	29.342	0.000
	Within Groups	99.375	23	4.973	-	-
	Overall	691.783	27	-	-	-

Table 5 displays statistically significant differences between groups after analysing Spacing (SP) and ONVG for 200 VM instances. Strong group variability is shown by the high F-values (28.734 for SP and 29.349 for ONVG), and the low p-values (0.000) verify that these differences are not the result of chance. The influence of ONVG (145.852) is more important than SP (0.133), according to the Mean Square values. According to our investigation, these elements are essential for maximizing performance and resource allocation in cloud computing settings.

4.5.2. Duncan's Multiple Range Test

The Duncan's Multiple Range Test is a statistical method for comparing the mean values of multiple sets (DMRT). Using numerical bounds determined by studentized range statistics, it categorizes variations among any more than two groups as significant or non-significant. The DMRT ranks the sets in either increasing or decreasing sequence, depending on the user's selection. DMRT was used for the post hoc analysis in order to identify the top-performing algorithm. Table 6 uses the DMRT to separate the method into three homogeneous groups.

Table 6 presents the statistical analysis of DMRT. The first homogeneous group indicates that the (Proposed) algorithm is executed as the equivalent of all other algorithms. FPO-NSGA-II, SDP, FFNN, SRE-HM, FC-BGSA, SGD-ML-DNN, DNN-TL and HNN-PSO are performing in the same table. To ascertain whether there is a substantial association between the algorithm outputs, the DMRT test is used in this research. The DMRT test was performed using a 95% significance level. Assume that $\alpha = 0.05$, the critical value is greater than the Sig value. The initial hypothesis (Ho) should be rejected and an alternate theory (H1) approved if there is a substantial difference between the given set of data. It allows post-hoc tests to be used, like DMRT, which was used in this case. If the sig value is greater than 0.05, however, post-hoc testing is not possible and the null assumption (Ho) can be accepted.

Table 6. Analysis of 200 VM Instances Using DMRT for ONVG.

Duncan's Multiple Range Test				
Methods	Number of instances (N)	Alpha subset =0.05		
		1	2	3
DGCN-SB-WSA (Proposed)	5	32.600	28.750	22.350
FPO-NSGA-II [16]	5	30.250	25.780	20.650
SDP [17]	5	28.320	24.890	19.750
FFNN [18]	5	27.500	23.600	18.920
SRE-HM [19]	5	26.750	22.950	18.470
FC-BGSA [20]	5	25.620	21.870	17.820
SGD-ML-DNN [21]	5	24.500	21.250	17.350
DNN-TL [22]	5	23.780	20.890	16.950
HNN-PSO [23]	5	22.900	20.100	16.520
Sig (p-value)	-	1.000	1.000	0.195

4.6. Ablation study for proposed approach

An ablation study for the proposed approach involves systematically removing or altering specific components to measure their individual contributions to the overall performance. For this approach, critical components include the DGCN, SBA and WSA. By testing the approach with and without each component, better understand the unique impact of each on metrics. Table 7 displays the ablation study results table.

Table 7. Ablation Study Results.

Model Variant	RU (%)	PC (kW)	CE (kg CO_2)	NAS	CF (kg)	RT (s)
Proposed (DGCN + SB-WSA)	98	27	180	175	300	23.1
Without DGCN	68	44	270	230	390	50.0
Without SBA	78	32	200	190	320	30.5
Without WSA	75	35	220	200	340	35.2
Without SB-WSA	70	40	250	220	370	42.0

The ablation study in table 7 demonstrates that performance deteriorates when essential components are removed. The optimal efficiency is attained by the whole model (98% RU, 27 kW PC, 23.1s RT). Response time, carbon emissions, and power usage all rise when DGCN, SBA, or WSA are removed. Without SB-WSA, the performance is poorest (40 kW PC, 42.0s RT), underlining how every element is required for the best results.

4.7. Discussion

The performance evaluation and ablation study present the efficiency of the proposed DGCN-SB-WSA technique in optimizing VM placement in cloud environments. For comparison research, the proposed model outperforms current approaches in crucial performance metrics such as carbon emission (180 kg CO_2), power consumption (27 kW), and resource utilization of 98%. Besides, it executes faster compared to others, with the fastest response time at 23.1 s, which ensures the accomplishment of effective work execution. Its environmental benefits are further reflected in the low number of active servers at 175 and carbon footprint at 300 kg. The ablation study confirms that every one of the three main components-DGCN, SBA, and WSA-plays its own role in the model's effectiveness. Its critical function in resource allocation is reflected by a steep increase in power consumption to 44 kW and response time to 50.0 s when DGCN is removed. Though the effect is marginally less severe than that caused by the removal of DGCN, the absence of SBA or WSA causes a surge in energy consumption and response time. The removal of both SBA and WSA leads to the highest degradation in performance, with an increased response time to 42.0 s and power consumption to 40 kW. This demonstrates how vital the combined functioning of SBA and WSA is for maximum resource utilisation and sustaining system stability. In general, results show that a whole model with all integrated components results in better performance, reflecting improved workload handling, reduced carbon emissions, and improved energy efficiency. Ensuring that resources are well utilized, waste is reduced, and operational delays are at a minimum, the proposed method promises the best possible VM placement.

5. Conclusion

To address the VM consolidation issues, an efficient Bio-VMP model is suggested in this study that considers the objectives of both cloud customers and service providers concurrently. The paper introduced the meta-heuristic optimization strategy SB-WSA for identifying optimal virtual machine allocation for both dynamic and static environments. After performing virtual machine placement, the proposed model achieved significant reductions in power usage, carbon emissions, and resource waste. When evaluated against existing techniques, the framework demonstrated notable improvements in key performance metrics, including carbon emissions at 180 kg CO_2, power consumption at 27 kW, and resource utilization at 98%.

Despite these promising results, certain limitations remain. The hybrid DGCN-SB-WSA framework introduces computational overhead due to its iterative dual-phase optimization and graph updating process, which may challenge real-time scalability in large-scale data centers. The model’s high-dimensional parameter space also demands considerable memory and processing power. Moreover, experiments were conducted using controlled workloads from the Google Cluster Dataset, which may not fully capture the heterogeneity of real-world cloud environments.

In the future, addressing these challenges with techniques such as distributed optimization, lightweight model design, and validation using diverse datasets like PlanetLab and hybrid-cloud traces will be pursued. Further extensions will also consider additional objectives in security, reliability, and trust to enhance robustness in practical cloud infrastructures.

All the Declarations and Statements

Author Contributions Statement

Viji Vinod – Conceptualization, Methodology, Writing - Original Draft.
V. J. Chakravarthy – Validation, Data Curation, Writing - Review & Editing.
N. Jayashri – Formal analysis, Writing - Review & Editing.
Bala Dhandayuthapani V. – Project administration, Writing - Review & Editing.

Shabeen Taj G. A. – Investigation, Resources, Software.
A. V. G. A. Marthanda: Supervision, Writing - Review & Editing.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement
The authors declare no conflicts of interest.

Funding Declaration
None.

Data Availability Statement
None.

Ethical Declarations
None.

Acknowledgments
None.

Declaration of Generative AI in Scholarly Writing
AI tools were used only for minor grammar checking and language refinement. All methodological development, analysis, and technical content were entirely carried out by the authors without AI assistance.

Abbreviations
The following abbreviations are used in this manuscript:

DGCN - Deformable Graph Convolutional Network
GCD - Google Cluster Dataset
NP - Nondeterministic Polynomial
PM - Physical Machine
QoS - Quality of Service
SBA - Swarm Bipolar Algorithm
SB-WSA - Swarm Bipolar with Walk-Spread Algorithm
VMP - Virtual Machine Placement
VMs - Virtual Machines

References

[1] Shanmugam, V., Ling, H.C., Gopal, L., Eswaran, S. and Chiong, C.W., 2024. Network-aware virtual machine placement using enriched butterfly optimisation algorithm in cloud computing paradigm. *Cluster Computing, vol. 27, no. 6, pp. 8557–8575, 2024.*
[2] Elsedimy, E.I., Herajy, M. and Abohashish, S.M., 2025. Energy and QoS-aware virtual machine placement approach for IaaS cloud datacenter. *Neural Computing and Applications, vol. 37, no. 4, pp. 2211–2237, 2025.*
[3] Çavdar, M.C., Korpeoglu, I. and Ulusoy, Ö., 2024. A utilization based genetic algorithm for virtual machine placement in cloud systems. *Computer Communications*, *214*, pp.136-148.
[4] Gopu, A., Thirugnanasambandam, K., AlGhamdi, A.S., Alshamrani, S.S., Maharajan, K. and Rashid, M., 2023. Energy-efficient virtual machine placement in distributed cloud using NSGA-III algorithm. *Journal of Cloud Computing*, *12*(1), p.124.
[5] Singh, A.K., Swain, S.R. and Lee, C.N., 2023. A metaheuristic virtual machine placement framework toward power efficiency of sustainable cloud environment. *Soft Computing*, *27*(7), pp.3817-3828.
[6] Sheeba, A. and Uma Maheswari, B., 2023. An efficient fault tolerance scheme based enhanced firefly optimization for virtual machine placement in cloud computing. *Concurrency and Computation: Practice and Experience*, *35*(7), p.e7610.
[7] Shi, F. and Lin, J., 2022. Virtual machine resource allocation optimization in cloud computing based on multiobjective genetic algorithm. *Computational Intelligence and Neuroscience*, *2022*(1), p.7873131.
[8] Balaji, K., Sai Kiran, P. and Sunil Kumar, M., 2023. Power aware virtual machine placement in IaaS cloud using discrete firefly algorithm. *Applied Nanoscience*, *13*(3), pp.2003-2011.
[9] Wang, N., Osmani, A. and Mirzaei, S., 2022. Dynamic placement of virtual machines using an improved multi-objective teaching-learning based optimization algorithm in cloud. *Transactions on Emerging Telecommunications Technologies*, *33*(9), p.e4529.
[10] Shirvani, M.H., 2023. An energy-efficient topology-aware virtual machine placement in cloud datacenters: a multi-objective discrete jaya optimization. *Sustainable Computing: Informatics and Systems*, *38*, p.100856.

[11] Nabavi, S.S., Gill, S.S., Xu, M., Masdari, M. and Garraghan, P., 2022. TRACTOR: Traffic-aware and power-efficient virtual machine placement in edge-cloud data centers using artificial bee colony optimization. *International Journal of Communication Systems*, *35*(1), p.e4747.
[12] Ghasemi, A., Toroghi Haghighat, A. and Keshavarzi, A., 2024. Enhancing virtual machine placement efficiency in cloud data centers: a hybrid approach using multi-objective reinforcement learning and clustering strategies. *Computing*, *106*(9), pp.2897-2922.
[13] Gabhane, J.P., Pathak, S. and Thakare, N., 2024. An improved multi-objective eagle algorithm for virtual machine placement in cloud environment. *Microsystem Technologies*, *30*(5), pp.489-501.
[14] Nabavi, S., Wen, L., Gill, S.S. and Xu, M., 2023. Seagull optimization algorithm based multi-objective VM placement in edge-cloud data centers. *Internet of Things and Cyber-Physical Systems*, *3*, pp.28-36.
[15] Belgacem, A., Mahmoudi, S. and Ferrag, M.A., 2023. A machine learning model for improving virtual machine migration in cloud computing. *The Journal of Supercomputing*, *79*(9), pp.9486-9508.
[16] Singh, A.K., Swain, S.R., Saxena, D. and Lee, C.N., 2023. A bio-inspired virtual machine placement toward sustainable cloud resource management. *IEEE Systems Journal*, *17*(3), pp.3894-3905.
[17] Karmakar, K., Banerjee, S., Das, R.K. and Khatua, S., 2022. Utilization aware and network I/O intensive virtual machine placement policies for cloud data center. *Journal of Network and Computer Applications*, *205*, p.103442.
[18] Swain, S.R., Parashar, A., Singh, A.K. and Lee, C.N., 2025. An intelligent virtual machine allocation optimization model for energy-efficient and reliable cloud environment. *The Journal of Supercomputing, 81(1), Art. no. 237, 2025.*
[19] Saxena, D. and Singh, A.K., 2022. A high availability management model based on VM significance ranking and resource estimation for cloud applications. *IEEE Transactions on Services Computing*, *16*(3), pp.1604-1615.
[20] Aghasi, A., Jamshidi, K. and Bohlooli, A., 2022. A thermal-aware energy-efficient virtual machine placement algorithm based on fuzzy controlled binary gravitational search algorithm (FC-BGSA). Cluster Computing, vol. 25, no. 2, pp. 1015–1033, 2022.
[21] Swain, S.R., Parashar, A., Singh, A.K. and Lee, C.N., 2024. Efficient straggler task management in cloud environment using stochastic gradient descent with momentum learning-driven neural networks. *Cluster Computing*, *27*(4), pp.4673-4685.
[22] Singh, G., Sengupta, P., Mehta, A. and Bedi, J., 2024. A feature extraction and time warping based neural expansion architecture for cloud resource usage forecasting. *Cluster Computing, vol. 27, no. 4, pp. 4963–4982, 2024.*
[23] Kumar, J., Saxena, D., Kumar, J., Bhadoria, A., Anjali, A. and Singh, A.K., 2025. A hybrid neural network and cooperative PSO model for dynamic cloud workloads prediction. Computing, 107(3), Art. no. 72, 2025.
[24] Park, J., Yoo, S., Park, J. and Kim, H.J., 2022, June. Deformable graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 36, No. 7, pp. 7949-7956).
[25] Kusuma, P.D. and Dinimaharawati, A., 2024. Swarm Bipolar Algorithm: A Metaheuristic Based on Polarization of Two Equal Size Sub Swarms. *International Journal of Intelligent Engineering & Systems, 17(2), pp. 377–389.*
[26] Kusuma, P.D. and Prasasti, A.L., 2023. Walk-Spread Algorithm: A Fast and Superior Stochastic Optimization. *International Journal of Intelligent Engineering & Systems, 16(5), pp. 275–288.*
[27] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis," in Proceedings of the Third ACM Symposium on Cloud Computing (SoCC), pp. 1–13, 2012.

Authors' Profiles

Viji Vinod is currently HOD Faculty of Computer Applications at Dr. M G.R. Educational and Research Institute in Maduravoyal, Chennai, Tamil Nadu 600095. She has published many research papers in many international journals and conferences. Her research area includes Computer science and Information sciences.

V. J. Chakravarthy is currently working as a Professor, Faculty of Computer Applications, Dr. M.G.R. Educational and Research Institute, Maduravoyal, Chennai, Tamil Nadu 600095, India. He has published many research papers in many international journals and conferences. His research area includes Computer science and Information sciences.

N. Jayashri is currently working as an Associate Professor, Faculty of Computer Applications, Dr. M.G.R. Educational & Research Institute, Maduravoyal, Chennai, Tamil Nadu 600095, India. She has published many research papers in many international journals and conferences. Her research area includes Computer science and Information sciences.

Bala Dhandayuthapani V. is currently with the Department of Information Technology, College of Computing and Information Sciences, University of Technology and Applied Sciences, Shinas campus, Oman. He has published many research papers in many international journals and conferences. His research area includes Computer science and Information security.

Shabeen Taj G. A. is currently working as an Assistant Professor, Department of Computer Science Engineering, Government Engineering College Ramanagar, B.M. Road, Ramanagara, Karnataka 562159, India. She has published many research papers in many international journals and conferences. Her research area includes Computer science, Information security and artificial intelligence techniques.

A. V. G. A. Marthanda is currently working as an Associate Professor, Department of Electrical and Electronics Engineering, Lakireddy Balireddy College of Engineering, Mylavaram, Andhra Pradesh, India. He has published many research papers in many international journals and conferences. His research area includes Computer science, Information security and artificial intelligence techniques.

How to cite this paper:

Viji Vinod, V. J. Chakravarthy, N. Jayashri, Bala Dhandayuthapani V., Shabeen Taj G. A., A. V. G. A. Marthanda, "Enhancing Energy Efficiency in Cloud Data Centers through Deformable Graph Convolutional Network-Aware Virtual Machine Placement with Hybrid Swarm Bipolar Walk-Spread Optimization", International Journal of Computer Network and Information Security(IJCNIS), Vol.18, No.4, pp. 187-209, 2026. DOI:10.5815/ijcnis.2026.04.10