Juxiang Hao[1]
Tingting Pang[1]
Guiyong Sheng[2*]
Qin Xu2, Jun Pan[2]

Resource Management using Multi-Scale Attention Convolutional Neural Networks in Containerized Cross-cloud Multi-cloud Environment

Journal of Electrical Systems

Abstract: - Cloud computing gives businesses rapid and easy access to a pool of dispersed, virtualized resources, likes virtual machines (VM) and containers, making them more competitive in the marketplace. They could also manage their business more skilfully. Even while the cloud makes it simple to install and manage business processes, it may be difficult to manage the highly variable resource requirements and guarantee the seamless execution of business activities in containerized multi-cloud scenarios. Hence, elastic resource provisioning is required to meet the demands of cloud providers, end users, manage over- and under-provisioning issues, and take QoS constraints, service level agreements (SLA) into account. In this manuscript, Resource management using Multi-Scale Attention Convolutional Neural Networks in containerized Cross-cloud multi-cloud environment (REMT-MCNN-CCMCE) is proposed for effective execution of business processes (BPs) in containerized multi cloud environment with guaranteed quality of service. The data for proposed REMT-MCNN-CCMCE method collected from GWA-T-12 bitbrains. Therefore, it is necessary to pre-process the workloads before considering them for more processing. Here workloads are filtered for abnormal, noisy interruptions with support of Regularized bias-aware ensemble Kalman filter (r-EnKF). This pre-processed data transferred to Self-Adaptative Multi-Kernel Clustering (SAMKC) for used to enable effectual scheduling of cloud workloads. In this work resource management is completed by using the Multi-scale Attention Convolutional Neural Networks (MACNN). This MACNN helps effective execution of BP's in containerized multi cloud environment with guaranteed QoS. The Sheep Flock Optimization Algorithm (SFOA) is utilized to suitable containers selection for dynamic cloud workloads. It enhances the performance of MACNN by optimizing its parameters, thereby increasing the effective execution of BP's in containerized multi cloud environment with guaranteed QoS. The proposed REMT-MCNN-CCMCE framework has been executed in Container Cloudsim platform with assessed utilizing performances likes SLA violation rate, CPU utilization, Response time, Execution cost, Energy consumption, Make‑span, Throughput. The REMT-MCNN-CCMCE method achieves 31.89%, 25.45% and 19.32% lower SLA violation rate, 32.12%, 23.49% and 30.94% higher CPU utilization and 26.87%, 34.65% and 23.94% lower response time when compared with the existing method's such as Multi-agent QoS-aware autonomic resource provisioning framework for elastic BPM in containerized multi-cloud environment (MAQ-ARP-EBPM-CMCE), Multi-cloud service provision depend on decision tree with two-layer Restricted Monte Carlo Tree Search (MCS-DTLR-MCTS) and Orchestration in Cloud-to-Things compute continuum: taxonomy, survey, future directions (OCT-CC-TSFD) respectively.

Keywords: Container Technology, Cloud Deployment Strategy,GWA-T-12 bitbrains, Multi-scale Attention Convolutional Neural Networks, Regularized bias-aware ensemble Kalman filter, Resource Management, Self-Adaptative Multi-kernel Clustering and Sheep Flock Optimization Algorithm.

I. INTRODUCTION

A growing paradigm for internet-based computing called "cloud computing" gives users access to variety of heterogeneous virtualized resources, including virtual machines (VMs) and containers housed in cloud data centers. The flexibility of the cloud and its capacity to dynamically allocate resources to changing workloads are its most important features. To meet the needs of cloud-based apps, cloud system is supposed to dynamically provide cloud resources depend on demand [1-3]. As a result, in order to rapidly create likes elastic resources, higher scalability, effective use, cloud providers must guarantee elasticity of cloud resources. This meets the goals of cloud users and providers by enabling the effective and affordable execution of cloud-based applications. Furthermore, cloud services may be effectively delivered to end users with guaranteed QoS by preserving flexibility in the cloud environment [4-6]. However, it's challenging to uphold BP's quality of service in a multi-cloud environment without violating SLAs. Furthermore, one of the main problems with the current cloud resource provisioning strategies is the equitable identification of the resources that must be provided for each service [7-9].Virtualization technology, which enables many users to share virtualized resources, is the

[1] [1]School of Traffic and Transportation, Xi'an Traffic Engineering Institute, Xi'an 710300, China

[2] Yangzhou Polytechnic Institutes, Yangzhou, Jiangsu 225127, China

[*]Corresponding author e-mail: m18952594816_1@163.com

Juxiang Hao: 18049086301@163.com

Tingting Pang: 13474038776@163.com"

fundamental component of cloud computing. Software or computational resources like virtual machines or containers can be utilized as these resources [10-12]. The introduction of virtualization based on containers offers the cloud system several advantages. Through the reduction of overhead, uniformity, efficiency, etc., it enhances overall performance [13, 14]. Containerization guarantees high performance and cost-effective elastic arrangement, managing BP. Moreover, it offers choices for resource auto-scaling depend on workload need, enabling effective execution of BP activities with assured QoS [15-17].

Assuring the resources needed to carry out corporate operations in multi-cloud computing is still significant problem, nevertheless. An under-provisioning issue arises when there are not enough resources supplied to match demand, which results in service delays that violate SLA [18, 19]. On the other hand, over-provisioning results in high computational costs when the resources provided exceed the need. Therefore, elastic and appropriate cloud resource provisioning may fulfil the demands of cloud-based application workloads by handling over- and under-provisioning issues, adhering to QoS limitations, cutting down on computational costs, and minimizing SLA violations.

The problem statement for this paper revolves around the challenges associated with ensuring quality of service (QoS) for business processes (BPs) in multi-cloud environment while avoiding SLA violations. In particular, the challenge is to manage the over- and under-provisioning of cloud resources to efficiently match dynamic workload needs. Even though containerization and virtualization technologies have benefits like increased scalability and performance, obtaining elastic and suitable resource provisioning is still quite difficult. To improve the execution of cloud-depend applications with assured QoS and minimal SLA violations, this article proposes techniques for effective resource allocation and management in multi-cloud systems [20-26].

Major contribution of this paper a follows;

- Resource management using Multi-Scale Attention Convolutional Neural Networks in containerized Cross-cloud multi-cloud environment (REMT-MCNN-CCMCE) is proposed.
- Utilizing the notion of autonomous computing inside the containerized multi-cloud setting to provide resource management elasticity for efficient BP execution while taking QoS and SLA constraints into account.
- During workload pre-processing, a partitioning approach is applied to filter out irregular and noisy interruptions from the workloads.
- Since business workloads vary in needs and are heterogeneous in nature, an initialization approach for centroid initialization is chosen to optimally cluster workloads into CPU intense, I/O intensive groups using Self-Adaptative Multi-Kernel Clustering (SAMKC). By reducing the possibility of resource shortage and waste, this assists in determining the quantity of resources that must be provided.
- Since BP has a dynamic workload need, one suggested approach is to use Multi-scale Attention Convolutional Neural Networks (MACNN), which operate iteratively to forecast future resource demand. This facilitates the management of fluctuating resource requirements and guarantees elastic resource allocation by adjusting cloud resources in a vertical or horizontal manner according to anticipated outcomes, so enabling seamless and effective task completion.
- Developed the SFOA to supply resources needed for the incoming workloads. Resource management performs better as a whole when workloads are assigned the best container according to several criteria and their resource needs.
- Formulating the QoS factors, including make-span, throughput, energy consumption, CPU usage, response time, execution cost, and SLA violation rate.
- Assessing the proposed structure by examining business workload traces with respect to SLA violation rate, CPU use, reaction time, cost of execution, energy usage, make-span, and throughput.

Remaining portion of this work structured below: section 2: literature review, section 3: defines proposed methodology section 4: illustrates results with discussion, section 5: conclusion.

II. LITERATURE REVIEW

Saif et.al [20], have presented, MAQ-ARP framework for EBPM in CMCE. This paper presents an effective methodology for multi-agent autonomic resource provisioning that guarantees QoS while enabling effectual execution of BPs in CMCE. Through demand forecasting and resource scaling for improved performance and elastic delivery, autonomous computing maximizes resource use. It uses local agents for K-means clustering of workloads, improved deep stacked auto-encoder (EDSAE) demand prediction, and container scaling. To

efficiently assign containers, global agents use multi-objective termite colony optimization. When the technique was utilized on Container Cloudsim with business workload traces, it performs noticeably better than alternatives in terms of make-span, throughput, cost, energy, CPU consumption, reaction time, and SLA violation rate.

Li et.al [21], have suggested, MCS provision depend on decision tree with two-layer Restricted MCTS. Here, presents multi-agent, distributed brokering technique for cloud service provisioning. Brokers can modify their resource introduction strategies depending on past performance by utilizing decision tree-depend user preference learning. Brokers use Restricted Monte Carlo Tree Search to quickly ascertain which cloud service was the best value for customers. Based on the Libcloud foundation, the model establishes a system of multi-cloud service connection, providing users with single interface to access best services available. The model outperforms conventional methods in meeting customer service demands, as shown by experimental results.

Ullah et.al [22], have suggested, OCT compute continuum: taxonomy, survey, future directions. To critically examine the landscape of existing research effort, this study attempts to compile investigation undertaken in orchestration for Cloud-to-Things continuum landscape, provide comprehensive taxonomy. In conclusion, this paper goes over the main issues that still need to be addressed and provide a conceptual framework derived from the analysis that was done. IoT devices are starting to become a need in this paper surroundings. Augmented reality, self-driving cars, smart cities, and smart manufacturing are just a few of the many fields in which the use of these technologies was expanding quickly. These Internet of Things use cases frequently ask for concurrent access to heterogeneous distant, local, multi-cloud compute resources in addition to widely dispersed geographic arrays of sensors.

Kumar and Anandahmala [23], have presented, Hash-depend message authentication code with Rijndael-depend multilevel security method for data storage in cloud environment. This investigation indicates that vendor lock-in may be significantly decreased with the appropriate use of innovative techniques. In order to minimize vendor lock-in, the solution offers a dynamic identity-based approach for multi-copy storage across many cloud servers. It provides security by storing multi-holomorphic verifiable tags on various servers and using the KCDH technique. Two SNP-based data centers for dynamic data storage techniques are used, along with data integrity audits and block hash merging. Utilizing the airport information for evaluation, the HMAC-Rijndael framework outperformed alternative encryption techniques, indicating improved security and effective cloud storage.

Wang et.al [24], have presented, Reproducible and Portable Big Data Analytics in Cloud. Here, open-source toolkit was developed facilitates: 1) fully automated E2E execution, replication with single command; 2) automated data and configuration storage for every execution; 3) customizable client modes according to user preferences; 4) querying the execution history;5) easy replication of previous executions in similar or different environments. This paper conducted thorough testing with four virtual CPU/GPU cluster-based big data analytics apps on AWS and Azure. This paper toolkit's capacity to deliver high execution performance, scalability, efficient reproducibility for cloud-depend big data analytics is demonstrated by the trials.

Ouchaou et.al [25], have suggested towards distributed saas management system in multi-cloud environment. To efficiently manage, store, retrieve cloud services inside federation, this work makes three contributions likes I a special cloud federation design, an appropriate service management scheme, service publication method. This approach combines a number of ideas, including the community deployment model, graph theory, clustering techniques, trust, and semantic Web (ontologies). This research goal was to improve user experience, increase profit, and automate the management process. These tests demonstrate how well the deployed management system and the suggested cloud federation architecture optimize storage capacity and promptly and effectively respond to user requests.

Yan and Sheng [26], have presented, Sdn+ K8s Routing Optimization Approach in 5G Cloud Edge Collaboration Scenario. The context of 5G cloud edge cooperation; this article suggests a new cloud native architecture type. Concurrently, it decouples, integrates context barriers with micro service mesh at several levels while optimizing SDN routing strategy at numerous levels. It does global route optimization depend on k8s + Sdn and multi threshold detection for underlying computational power, network form. Experiments validate the progressiveness of route optimization, stability, dependability of 5g + cloud edge collaboration architecture.

III. PROPOSED METHODOLOGY

In this section, Resource management using Multi-Scale Attention Convolutional Neural Networks in containerized Cross-cloud multi-cloud environment is discussed. Pre-processing, task clustering, resource management, and container selection are all covered in the proposed methodology section dataset. Applied to a containerized cross-cloud multi-cloud environment, the inter-sequence analysis technique has many potential uses and may dramatically cut costs and boost efficiency. In addition to being a helpful technique for improving model practicability, the proposed REMT-MCNN-CCMCE prediction model seeks to boost the effective execution of BPs in containerized multi-cloud environment by assured QoS accuracy. Block diagram of proposed REMT-MCNN-CCMCE is given below in figure 1,

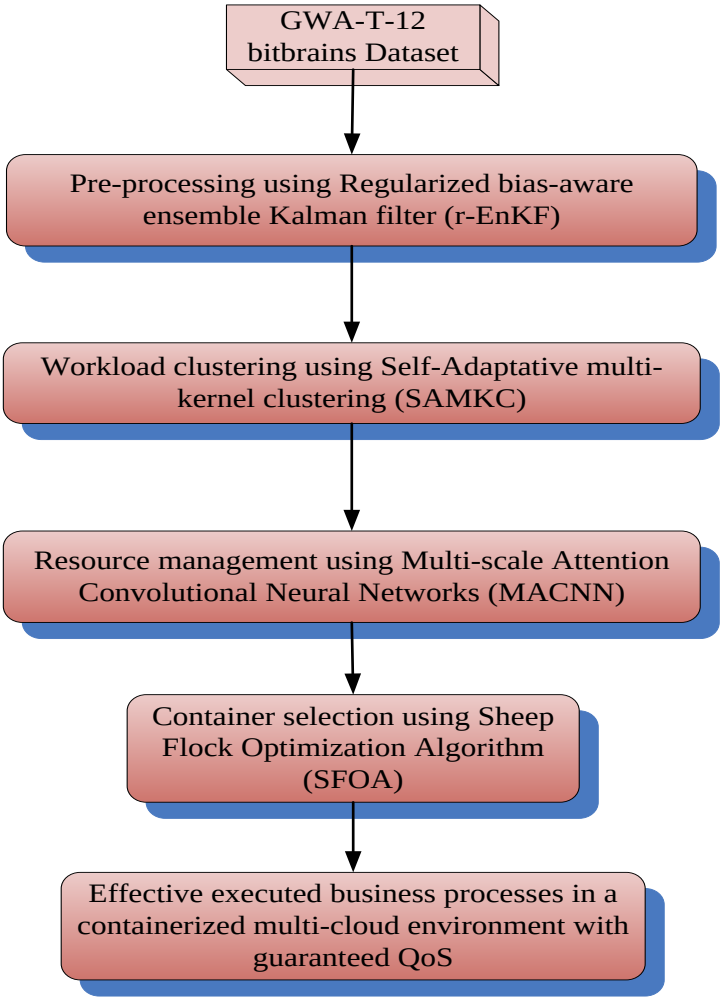

Figure 1: Block diagram of proposed REMT-MCNN-CCMCE

A. Data Acquisition

In this section the GWA-T-12 bitbrains Dataset [27] is discussed. The GWA-T-12 bitbrains dataset includes performance characteristics of 1,750 VMs from dispersed datacentre owned by Bitbrains, a business computing and managed hosting company. Numerous large banks (ING), credit card companies (ICS), insurers (Aegon), etc. are among customers. Programs utilized in the solvency domain are hosted by Bitbrains; providers of programs include Algorithmic and Towers Watson. Financial reporting is the usual purpose for these apps, and it happens mostly at the conclusion of the fiscal quarter. All file comprises performance metrics of VM. Such files are organized according by traces: fast Storage, Rnd. 1,250 virtual machines linked to fast storage area network storage devices make up first trace, called fast Storage. The 500 virtual machines (VMs) in the second trace, Rnd, are either linked to faster SAN devices or to the significantly slower Network Attached Storage devices. Because storage linked to fast storage machines performs better than that of the Rnd trace, a larger percentage of application servers, compute nodes are included in fast Storage trace. On the other hand, this

dataset see a larger percentage of management machines for the Rnd trace, which just need less-performing storage that is used less frequently.

B. Pre-processing using Regularized bias-aware ensemble Kalman filter

In this section, r-EnKF [28] is discussed. The r-EnKF gathered the data from GWA-T-12 bitbrains Dataset for pre-processing. Workloads should be pre-processed before being thought about for additional processing. Here, the workloads are screened to remove unnatural and distracting disruptions. This method determines if an unbiased analysis is "good" in a biased model situation if unbiased state (i) matches the truth, (ii) has a small-norm bias, demonstrated. This r-EnKF is based on the assumption that the model is qualitatively correct, that is, that it captures primary physical mechanisms required to give evaluation that is consistent by investigational evidence. Additionally, the r-EnKF approach makes the assumption that this method can identify a set of model parameters that will reduce the model bias's norm. In light of this, this r-EnKF method develops a data assimilation problem that minimizes the estimator's bias in addition to the estimation's uncertainty and distance from the observables. The problem that equation (1) shows is mathematically presented by this r-EnKF.

$$L(\psi_l) = \left\|\psi_l - \psi_l^h\right\|^2_{E^{h-1}_{\psi\psi}} + \left\|a_l - f_l\right\|^2_{E^{-1}_{ff}} + \gamma\left\|d_l\right\|^2_{E^{-1}_{dd}}' \ for\ l = 0,....,o-1 \quad (1)$$

Here, $\left\|\cdot\right\|^2_{E^{-1}}$ represent the operator, E^{-1} represent the semi-positive define matrix, γ denotes the hyper parameter, ψ_l denotes the analysis state and l denotes the ensemble number. If, for every ensemble member, an analysis state minimizes the regularized bias conscious cost function. This is seen in equation (2).

$$\frac{1}{2}\frac{fL}{f\psi_l}\bigg|_{\psi_l^c} = E^{g-1}_{\psi\psi}\left(\psi_l^c - \psi_l^h\right) + \frac{fa_l}{f\psi_l}\bigg|^V_{\psi l_l^c} E^{-1}_{ff}\left(a_l^c - f_l\right) + \gamma\frac{dd_l}{f\psi_l}\bigg|^V_{\psi_l^c} E^{-1}_{dd} d_l^c = 0 \quad (2)$$

Here, $\left\|\cdot\right\|^2_{E^{-1}}$ represent the operator, E^{-1} represent the semi-positive define matrix, γ denotes the hyper parameter, f denotes the noisy data, ψ_l denotes the analysis state and l denotes the ensemble number. This technique assumes that analysis state is close enough to forecast to allow r-EnKF to linearize the analysis bias, as in equation (3), in order to solve mathematically.

$$d_l^c \approx d_l^h + L^h O\left(\psi_l^c - \psi_l^h\right) \quad (3)$$

Here, d_l^h represent the bias, L^h represent the jacobian, O denotes the state, ψ_l denotes the analysis state and l denotes the ensemble number. It is significant that the minimization issue becomes quadratic as a result of linearization. The (r-EnKF) is displayed in equation (4) by grouping the variables in the analysis state and using the Woodbury matrix inversion formula.

$$\psi_l^c = \psi_l^h + M\left[\left(\mathrm{I} + L^h\right)\left(f_l - a_l^h\right) - \gamma E_{ff} E^{-1}_{dd} L^h d_l^h\right] \quad (4)$$

Here, d_l^h represent the bias, L^h represent the jacobian, f denotes the noisy data, γ denotes the hyper parameter, ψ_l denotes the analysis state and l denotes the ensemble number. The regularized Kalman gain matrix is located at equation (5),

$$M = E^h_{\psi\psi} O^U\left[E_{ff} + \left(\mathrm{I} + L^h\right) O C^h_{\psi\psi} O^U\left(\mathrm{I} + L^h\right)^U + \gamma E_{ff} E^{-1}_{dd} L^h O E^h_{\psi\psi} O^U L^{h^U}\right]^{-1} \quad (5)$$

Here, M represent the regularized Kalman gain matrix, E^{-1} represent the semi-positive define matrix, ψ_l denotes the analysis state, L^h represent the jacobian, O denotes the state and γ denotes the hyper parameter. By employing this technique, r-EnKF pre-processes the collected data to remove anomalous and disruptive disruptions, hence enhancing resource management effectiveness and maximizing cross-cloud multi-cloud deployment in container technology.

C. Workload clustering using Self-Adaptative Multi-Kernel Clustering

The SAMKC [29] method of clustering is covered in this section. SAMKC's preprocessing of data makes cloud workload scheduling possible. This SAMKC aids in figuring out how much resource needs to be supplied by

lowering the likelihood of waste and resource scarcity. Each mode discrete state in SAMKC is associated with a cluster and a multi-kernel regression function. Thus, given this equation (6), it follows.

$$\forall\left(\chi_y^v,\phi_y^u\right)\in\xi^v\times \mathrm{X}^v,\chi_y^v=\left\{Q_v\in\Pi/\phi_y^v\left(z_v\right)\geq 0\right\} \quad (6)$$

Here, χ_y^v represent cluster, ϕ_y^u represent the multi-kernel regression function, Π denotes the data space and v denotes the time. In this instance, all of the cluster data that is currently accessible in the data space is used to train the multi-kernel regression function. The kernel expansion with a gap defines the multi-kernel regression function at a time. It is shown in equation (7),

$$\phi_y^v\left(z_v\right)=\sum_{b=1}^{R_y}\sum_{k=1}^{f}\left(r_{b,y}^v-r_{b,y}^{*v}\right)\eta\left(z_{k,b},z_{k,v}\right)+\delta_y^v \quad (7)$$

Here, ϕ_y^u represent the multi-kernel regression function, v denotes the time, b denotes the support vector, $r_{b,y}^v$ and $r_{b,y}^{*v}$ are represent the lagrangian coefficient and δ_y^v represent the gap. This approach defines the kernel function in a Hilbert space as given by equation (8)

$$\left\langle\Phi\left(z_1\right),\Phi\left(z_2\right)\right\rangle_M=\sum_{k=1}^{f}\eta_k\left(z_{k,1},z_{k,2}\right) \quad (8)$$

Here, $\left\langle .,.\right\rangle_M$ represent scalar product in Hilbert Space and z denotes cluster. Prior to assessing the similarity measure, look for the class's winning support vector. This equation (9) is therefore provided for it.

$$U=\arg\min_{k=1,\ldots P_y}\left\|\Phi\left(z(v)\right)-\Phi\left(\Omega_{k,y}^v\right)\right\| \quad (9)$$

Here, $\Omega_{k,y}^v$ represent the i-th node of the cluster, z denotes the cluster and v denotes the time. The metrics of similarity described in equation (10)

$$\tau\Phi_{v,y}=\frac{\varepsilon}{2}\times\left\|\Phi\left(z(v)\right)-\Phi\left(\Omega_{U,y}^v\right)\right\|=\varepsilon\sqrt{1-g^{-d\left\|z(v)-\Omega_{U,y}^v\right\|2}}=\varepsilon\sqrt{1-g^{-d\left\|z(v)-\Omega_{U,y}^v\right\|2}} \quad (10)$$

Here, $\Omega_{U,y}^v$ represent the similarity measures, z denotes the cluster and v denotes the time. Ultimately, using the equation (11) this strategy can provide a set of winning clusters.

$$\xi^U=\left\{\chi_y^v/\chi_y^v\in\xi^v,\tau\Phi_{v,y}\geq\lambda_{vj}\right\} \quad (11)$$

Here, ξ^U represent the set of winning cluster, χ_y^v represent the cluster and λ_{vj} denotes the acceptable threshold. Alongside the discrete state estimate process, traffic data allocation is executed. By doing so, cardinality of winning clusters may be used to construct learning rules of SAKM.

D. Resource management using Multi-scale Attention Convolutional Neural Networks

In this section, MACNN [30] is discussed. Resource management is major task in cloud. Using MACNN for resource management, the suggested REMT-MCNN-CCMCE approach helps attaining elasticity of dynamic business processes to satisfy QoS necessities. Utilizing the control loop, which consists of the main elements like monitoring, analyzing, planning, and execution, is the resource management algorithm. The monitored data from monitor phase is stored for later usage using the knowledge base. The MACNN architecture is divided into four components. The MA layer, which makes up the first two levels, is fundamental layer of entire architecture. An MA module in the MA layer uses MAM to determine significance of all feature map produced by multi-scale convolution. The output portion, which provides the categorization result, is the last component. A technique called the Multi-scale Attention Mechanism uses the relative relevance of all feature map produced by multi-scale convolution to highlight and suppress less valuable feature maps. Resource management improvement is the MAM's main objective. The attention block adds channel-wise weight to emphasis on key feature maps, improving recognition capacity for classification. The multi-scale block generates multiple widths of receptive field to collect different scales of temporal information. Convolutional layer is the first layer in a multi-scale block. It extracts features and creates feature maps by applying a convolution operation with shared

weights to the input. Global average pooling layer is first layer in the attention block. It computes mean of each values for all feature map, condensing global temporal information into channel descriptor and producing channel-wise statistics. Multi-scale block's output is also the attention block's input. Let's look at the channel-wise statistics that are produced by reducing the feature map length; this gives us equation (12).

$$B_p = \frac{1}{N_h}\sum_{k=1}^{N_h} z_p(k) \tag{12}$$

Here, B_p represent the channel-wise statistics, N_h represent the feature map length and z_p denotes the multi-scale block. The channels of second completely connected layer have a sigmoid activation function, while the channels of the first layer of first fully connected layer have activation function of channels, where equation (13) yields the weight's p-th element.

$$U_p = \sigma\left(Y_2\delta\left(Y_1 b_p\right)\right) \tag{13}$$

Here, U_p represent the weight of p-th element, Y_1 and Y_2 represent the parameters of two fully connected layers, δ denotes the ReLU activation function and σ denotes the activation function. Applying channel-wise weight to feature maps yields attention block's output, and equation (14) computes the attention block's p-th element.

$$\tilde{z}_p = U_p \cdot z_p \tag{14}$$

Here, $\tilde{z}_p$ represent the attention block of p-th element, U_p represent the weight and z_p denotes the multi-scale block. The resource management function of the cloud major may be completed by this MACNN. With assured QoS, this MACNN facilitates effectual execution of BPs in containerized multi-cloud context.

E. Container selection using Sheep Flock Optimization Algorithm (SFOA)

In this section, SFOA [31] is discussed. This proposed REMT-MCNN-CCMCE approach improves effectual execution of business processes in containerized multi-cloud environment assured QoS by using SFOA for container selection. Sheep are ruminant, four-legged mammals that are mostly raised as agricultural animals. From a conceptual perspective, a sheep searches a nearby area after being given pasture; if it finds a better spot to graze, it will relocate; sheep hunt for the most profit. The goal of the sheep herd's behavior is to transfer the flock to the maximum location, as global maximum place may be beyond sheep's grazing radius. A sheep's desire to stay in its own grazing area is its primary consideration. Three other factors also influence location of the sheep: shepherd's order; sheep's interest in moving to the best place for past experiences; sheep's interest in approaching other sheep. SFOA is divided into dual sections: grazing section, move section. The first factor falls under first part, while the other three factors fall under second. In addition to sheep, goats can migrate in different ways within herds.

Step 1: Initialization

The population size, maximum iteration, boundaries, cost function, dimensions, maximum velocity are defined as starting variables in this procedure. The quantity of sheep, goats in pastures is population size. The maximum iteration refers to the quantity of optimization iterations. The greatest and lowest values that any variable can have are its bounds. One function gives value of input variables is cost function. The quantity of problem dimensions is known as dimensions. Max velocity may be calculated using equation (15).

$$VMax = 0.1 * (ub - lb) \tag{15}$$

Here, $VMax$ represent the maximum velocity; ub & lb represent upper bound, lower bound of all variable.

Step 2: Random generation

The weight parameters are formed randomly generated. The values generated randomly between 0 and 1.

Step 3: Fitness function

Fitness function creates random solution form initialized values. It calculated using optimizing parameter. Thus it is shown in equation (16),

$$Fitness\ \ Function = optimizing\left\lfloor U_p\ and\ \tilde{z}_p \right\rfloor \tag{16}$$

Step 4: Exploration phase

During the SFOA exploration phase, multimodal functions exhibit a greater number of local optimums than unmodal functions, with the number of these optimums increasing exponentially with issue size. These test problems are highly beneficial for assessing an optimization algorithm's exploration capacity. Equation (17) calculates movement resulting from a sheep's desire in approaching other sheep.

$$x_{other,1} = E * Ran(1, Dimen) * (Z_{RandomSheep} - Z) \quad (17)$$

Here, Z represent the current sheep position, $Z_{RandomSheep}$ represent the random sheep position, $Ran(1, Dimen)$ denotes the random array between $0 \& 1$ and $x_{other,1}$ denotes the sheep's interest speed.Equation (18) calculates the movement resulting from the goat's desire in the best experience from earlier.

$$x_{Lbest,2} = (1-U) * 2 * Ran(1, Dimen) * (Z_{Lbest} - Z) \quad (18)$$

Here, Z represent the current sheep position, x_{Lbest} denotes best fitness discovered by current goat, $Ran(1, Dimen)$ denotes the random array between $0 \& 1$ and $x_{Lbest,2}$ denotes the sheep interest in moving towards.

Step 5: Exploitation

SFOA's exploitation phase involves looking for superior solutions around the one that is currently in place. There is some competition between these two phases since if this method focuses too much on one; this method won't have enough time for the other. These functions enable assessing the algorithm's potential for exploitation. Equation (19) computes the movement resulting from shepherd's order to the optimal position.

$$x_{sh2,2} = (1-U) * 2 * Ran(1, Dimen) * Z_{GBest} - Z \quad (19)$$

Here, Z represent current sheep position, Z_{GBest} represent best fit until now and $x_{sh2,2}$ denotes the shepherd's order speed. Equation (20) states that the computed speed is gathered from the present position to determine the position of the following sheep.

$$z_{(Iter+1)} = z_{(Iter)} + X_{o,2} \quad (20)$$

Here, $z_{(Iter)}$ represent the position of each member in iteration and $X_{o,2}$ denotes the speed. The proposed REMT-MCNN-CCMCE approach uses the SFOA optimization strategy to choose appropriate containers for workloads in the dynamic cloud. With assured QoS, this approach enhances effectual execution of business processes in containerized multi-cloud environment.

Step 6: Termination

In the MACNN, the weight parameters for generators are optimized using the SFOA, dynamically adjusting weights inspired by celestial mechanics. The iterative refinement, guided by halting criteria $VMax = VMax + 1$, ensures optimal weight convergence, maximizing MACNN's generator performance. Then flowchart of SFOA for optimizing the weight parameters of MACNN for enhances effective implementation of BP's in containerized multi cloud environment by guaranteed QoS.Figure 2 showsflowchart of SFOA for optimizing the weight parameters of MACNN for enhances effective implementation of BP's in containerized multi cloud environment with guaranteed QoS.

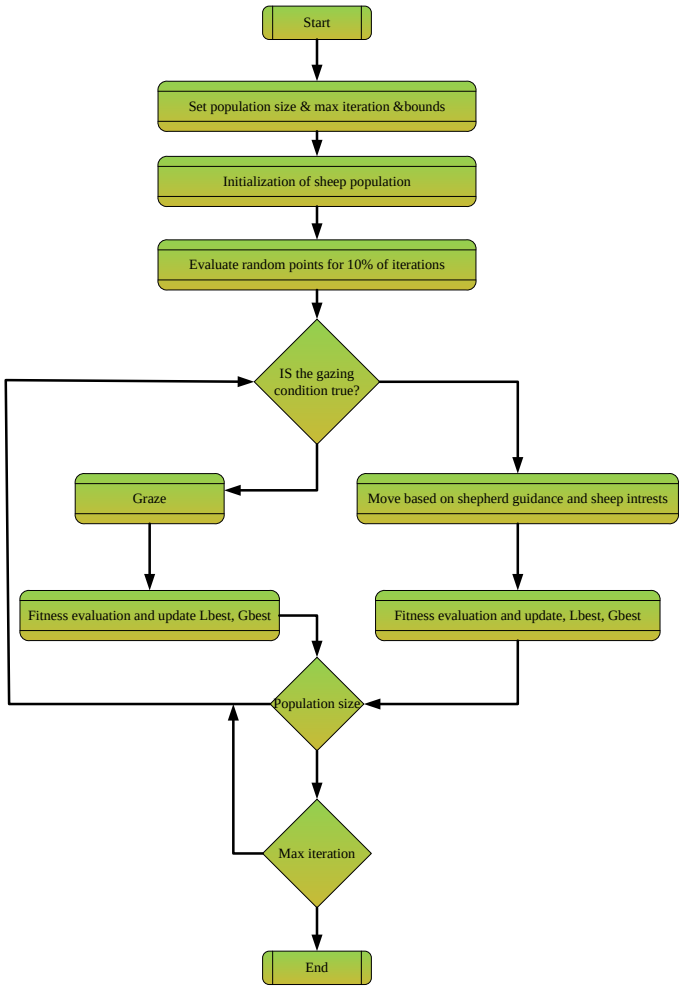

Figure 2: flowchart of SFOA for optimizing the weight parameters of MACNN for enhances effective implementation of BP's in containerized multi cloud environment with guaranteed QoS.

IV. RESULT AND DISCUSSION

In this paper, Resource management using Multi-Scale Attention Convolutional Neural Networks in containerized Cross-cloud multi-cloud environment is discussed. The proposed REMT-MCNN-CCMCE technique is executed in Container Cloudsim platform,assessed by utilizing several performance metrics SLA violation rate, CPU utilization, response time, execution cost, energy consumption, make‑span, throughput. The result of REMT-MCNN-CCMCE approaches was compared with existing MAQ-ARP-EBPM-CMCE, MCS-DTLR-MCTS and OCT-CC-TSFD techniques.

A. Performance measures

This is a crucial step for determining the exploration of optimization algorithm. Performance measures to evaluate to access performance likes SLA violation rate, CPU utilization, response time, execution cost, energy consumption, make‑span, Throughput.

1) SLA violation rate

The frequency or proportion of times the service provider fails to fulfil the agreed-upon SLA commitments is measured by the Service Level Agreement (SLA) violation rate. As may be seen in equation (21),

$$SLA\ violation\ rate = \left(\frac{N}{T}\right) \times 100\% \qquad (21)$$

Here, N denotes number of SLA violations, T denotes total number of SLA instances.

2) CPU utilization

The amount of CPU's processing power that is being utilized at any particular time is measured by CPU utilization. A percentage is frequently used to express it. The definition of the CPU usage formula is given in equation (22).

$$CPU\ utilization = \left(\frac{T-I}{T}\right)\times 100\% \quad (22)$$

Here, T denotes the total CPU time and I denotes the idle CPU time.

3) Response time

Response time, often known as latency, is a measurement of how long it takes system to react to request. Equation (23), which represents the response time formula,

$$\text{Re}sponse\ time = C + N + S \quad (23)$$

Here, C represent the client processing time, N represent the network transmit time and S denotes the server processing time.

4) Execution cost

The difference between a security's execution price and the price that would have existed in the absence of a trade; this difference can be further broken down into expenses related to market timing and effect.

5) Energy consumption

Energy consumption is the process of using a system's supply of power or energy. Watts, Giga Joules, and kilograms of oil equivalent annually (kg/a) are used to measure consumption. Equation (24) illustrates it.

$$C = Q \times h/1000 \quad (24)$$

Here, C represent the energy, Q represent the power and h denotes the hours.

6) Make-span

"Make-span" is a term used in computer science and parallel computing to describe overall time required to finish group of activities when they are run concurrently on several processors or resources.

7) Throughput

The number of units of work finished or processed in a certain amount of time is called throughput. It is frequently used to quantify the rate of data transfer, processing speed, or production output in a variety of applications, including networking, computing, and manufacturing. Equation (25) illustrates it.

$$Throughput = \frac{W}{t} \quad (25)$$

Here, W represent the total work completed and t represent the time taken for work.

B. Performance analysis

Figure (3 to 9) portrays the simulation outcomes of REMT-MCNN-CCMCE method proposed. Proposed REMT-MCNN-CCMCE method is compared with existing as MAQ-ARP-EBPM-CMCE, MCS-DTLR-MCTS and OCT-CC-TSFD methods.

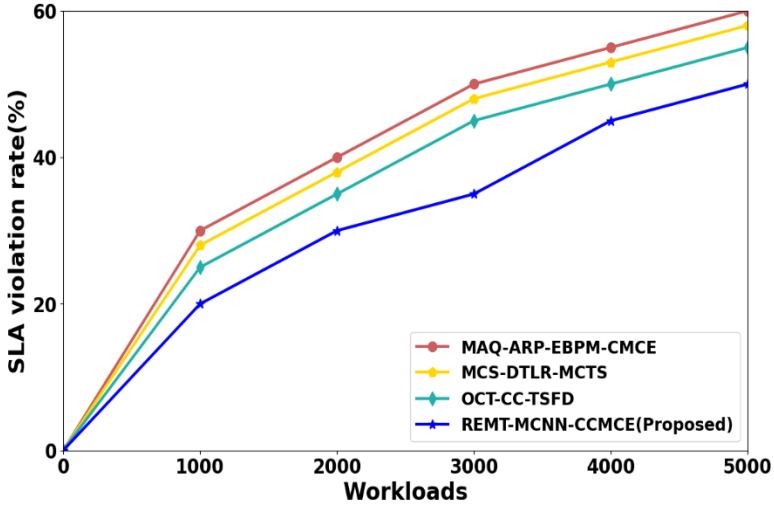

Figure 3: SLA violation rate analysis

Figure 3 depicts the analysis of SLA violation rate. Here; proposed REMT-MCNN-CCMCE technique attains 33.89%, 28.43% and 25.32% lower SLA violation rate for execution of BP's in a containerized multi cloud environment as analyzed with existing MAQ-ARP-EBPM-CMCE, MCS-DTLR-MCTS and OCT-CC-TSFD methods respectively.

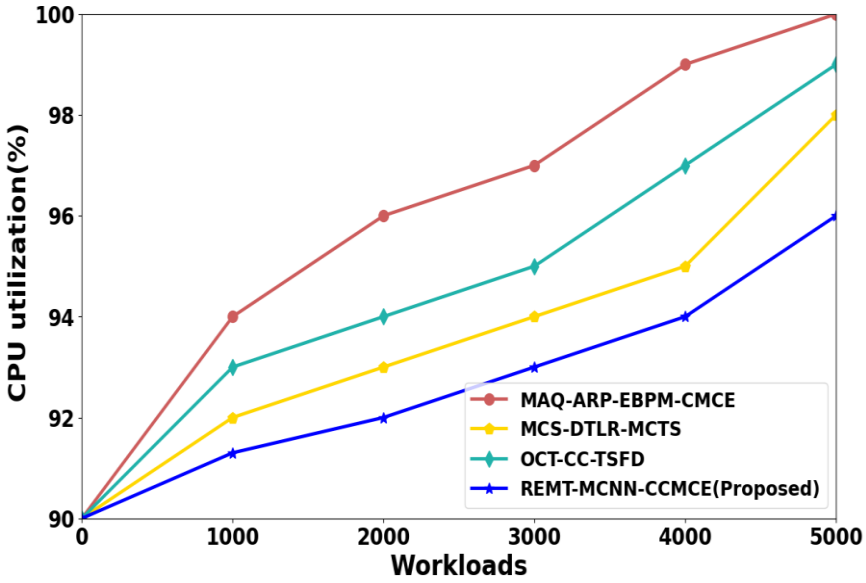

Figure 4: Performance analysis of CPU utilization

Figure 4 depicts the analysis of CPU utilization. Here; proposed REMT-MCNN-CCMCE technique attains 34.97%, 28.13% and 18.89% lower CPU utilization for execution of BP's in a containerized multi cloud environment as analyzed with existing MAQ-ARP-EBPM-CMCE, MCS-DTLR-MCTS and OCT-CC-TSFD methods respectively.

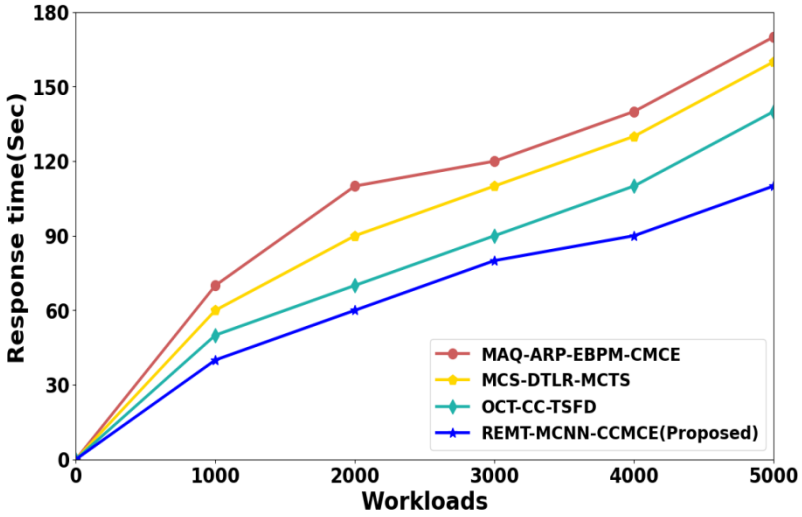

Figure 5: Performance analysis of Response time

Figure 5 depicts the analysis of Response time. Here; proposed REMT-MCNN-CCMCE technique attains 19.77%, 23.56% and 28.42% lower response time for execution of BP's in a containerized multi cloud environment as analyzed with existing MAQ-ARP-EBPM-CMCE, MCS-DTLR-MCTS and OCT-CC-TSFD methods respectively.

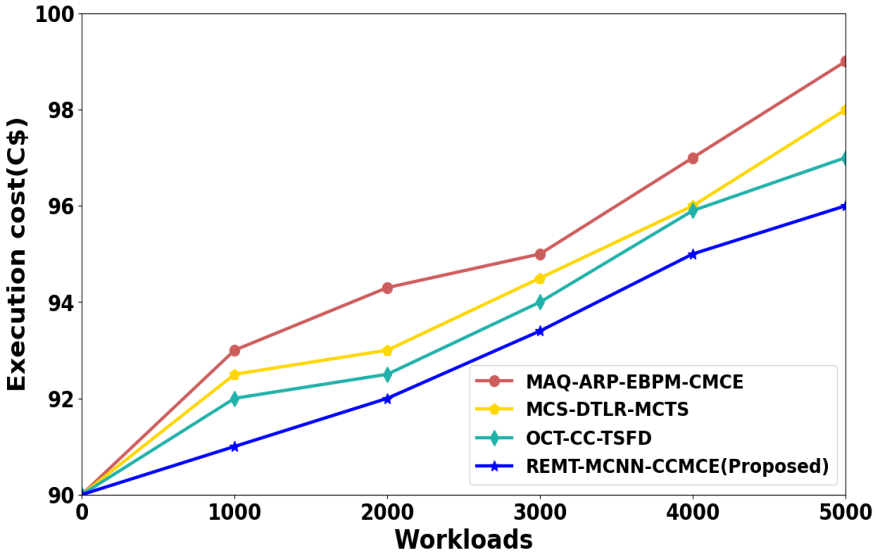

Figure 6 depicts the analysis of Execution cost

Figure 6 depicts the analysis of Execution cost. Here; proposed REMT-MCNN-CCMCE technique attains 34.75%, 17.41% and 19.63% lower execution cost for execution of BP's in a containerized multi cloud environment as analyzed with existing MAQ-ARP-EBPM-CMCE, MCS-DTLR-MCTS and OCT-CC-TSFD methods respectively.

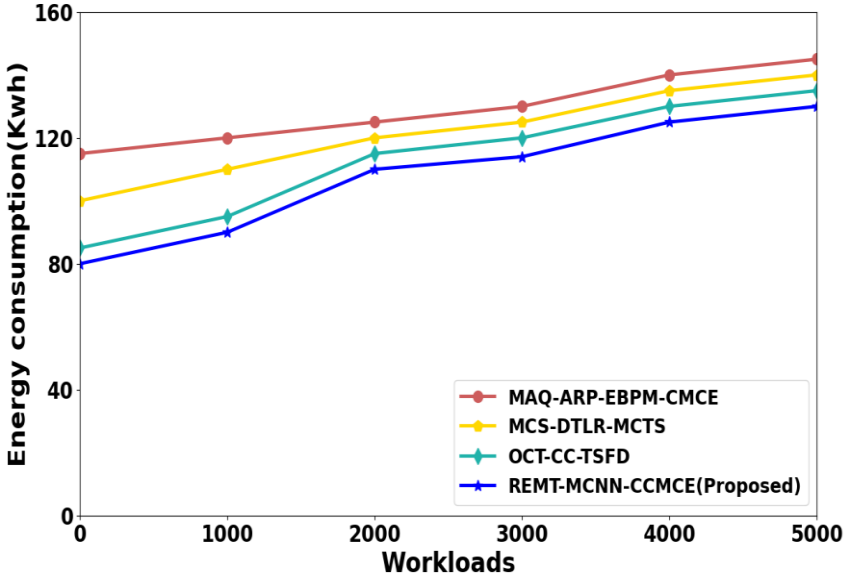

Figure 7: Performance analysis of Energy consumption

Figure 7 depicts the analysis of energy consumption. Here, proposed REMT-MCNN-CCMCE technique attains 34.75%, 26.41% and 20.63% lower energy consumption for execution of BP's in a containerized multi cloud environment as analyzed with existing MAQ-ARP-EBPM-CMCE, MCS-DTLR-MCTS and OCT-CC-TSFD methods respectively.

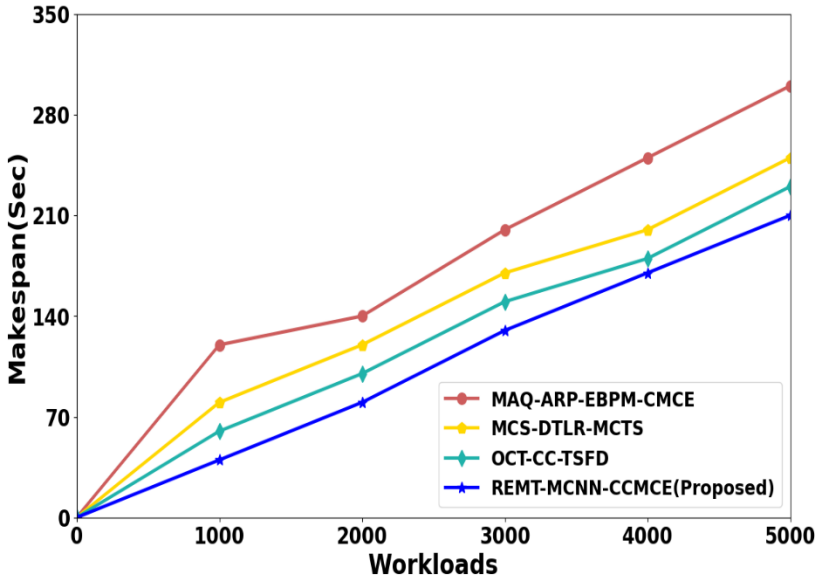

Figure 8: Performance analysis of Make-span

Figure 8 depicts the analysis of Make-span. Here, proposed REMT-MCNN-CCMCE technique attains 34.68%, 29.84% and 24.76% lower make-span for execution of BP's in a containerized multi cloud environment as analyzed with existing MAQ-ARP-EBPM-CMCE, MCS-DTLR-MCTS and OCT-CC-TSFD methods respectively.

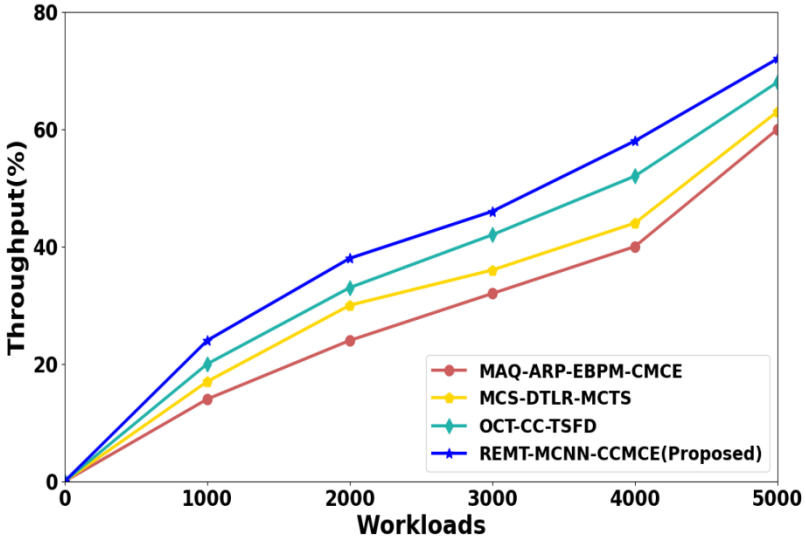

Figure 9: Performance analysis of throughput

Figure 9 depicts the analysis of throughput. Here; proposed REMT-MCNN-CCMCE technique attains 32.82%, 23.91% and 16.78% higher throughput for execution of BP's in a containerized multi cloud environment as analyzed with existing MAQ-ARP-EBPM-CMCE, MCS-DTLR-MCTS and OCT-CC-TSFD methods respectively.

C. Discussion

The information gathered for the proposed REMT-MCNN-CCMCE approach using GWA-T-12 bitbrains. Therefore, before assessing the workloads for further processing, pre-processing is required. Here, an ensemble Kalman filter that is aware of bias and regularizes it (r-EnKF) is used to filter the workloads for anomalous and noisy interruptions. Self-Adaptative Multi-Kernel Clustering (SAMKC) received this pre-processed data in order to schedule cloud workloads more effectively. In this research, Multi-scale Attention Convolutional Neural Networks (MACNN) are used to manage resources. With assured QoS, the MACNN facilitates effectual execution of BPs in containerized multi-cloud context. For dynamic cloud workloads, the SFOA is utilized to pick containers that are appropriate. Through parameter optimization, MACNN's performance is improved, leading to more effective BP execution in containerized multi-cloud environment by assured QoS. In instance of

result, the average highest outcomes of the approach were compared to average highest results in existing methods like MAQ-ARP-EBPM-CMCE, MCS-DTLR-MCTS and OCT-CC-TSFD respectively. The accuracy values of REMT-MCNN-CCMCE MAQ-ARP-EBPM-CMCE, MCS-DTLR-MCTS and OCT-CC-TSFD are lower than proposed technique. The proposed framework an average throughput of 99.94% analysed with throughput of 92.82% for the comparison approaches. Reduced execution cost of 97.91% and reduced energy usage of 96.53% as comparison to the existing approaches. The proposed method REMT-MCNN-CCMCE has high throughput and lower response time evaluation metrics than existing methods. As a result, the suggested methodology is less expensive than the compared techniques. Consequently, the proposed approach executes BPs more effectively and efficiently in containerized multi-cloud environment by assured QoS.

V. CONCLUSION

In this section, Resource management using Multi-Scale Attention Convolutional Neural Networks in containerized Cross-cloud multi-cloud environment was successfully implemented. The QoS needs of cloud users are impacted by dynamic nature of multi-cloud environment and dynamic submission of complicated, changing workloads from business partners at varying intervals. To meet needs of cloud users, providers while preventing under- and over provisioning of resources in multi-cloud environments, an effective and elastic resource provisioning system is required. This proposed REMT-MCNN-CCMCE method proposes an effective MACNN resource management paradigm that takes QoS restrictions into account when allocating resources to incoming BP workloads. The global agent repeatedly runs the SFOA algorithm resource provisioning framework to provide dynamic workloads into containers according to resource needs. In this step, key parameters are tuned to guarantee that workloads on the provided containers are scheduled as efficiently as possible. With average improvement rate of 57.3% for SLA violation rate, 24.67% for CPU utilization, 11.40% for response time, 15.49% for execution cost, 20.56% for energy consumption, 32.47% for make span, 54.52% for throughput, simulation, investigational analysis demonstrated that proposed method outperformed existing methods in each considered metrics. This method concludes that the presented architecture is a viable solution for managing diverse BP workloads in a multi-cloud context, as well as varying resource needs. The goal is to use this methodology in the future by utilizing dynamic and real-time workload traces from various cloud platforms. In order to provide elastic resource provisioning, it is also beneficial to take big data from the data warehouse into account due to the enormous rise in data rates.

Acknowledgement

1.Jiangsu University Philosophy and Social Science Research Project (2021SJA2019)
2.Qinglan Engineering Project of Jiangsu

REFERENCES

[1] Vhatkar, K.N., & Bhole, G.P. (2022). Optimal container resource allocation in cloud architecture: A new hybrid model. *Journal of King Saud University-Computer andInformation Sciences, 34*(5), 1906-1918.

[2] Ahmad, I., AlFailakawi, M.G., AlMutawa, A., & Alsalman, L. (2022). Container scheduling techniques: A survey and assessment. *Journal of King Saud University-Computer and Information Sciences, 34*(7), 3934-3947.

[3] Baburao, D., Pavankumar, T., & Prabhu, C.S.R. (2023). Load balancing in the fog nodes using particle swarm optimization-based enhanced dynamic resource allocation method. *Applied Nanoscience, 13*(2), 1045-1054.

[4] Lakhan, A., Memon, M.S., Mastoi, Q.U.A., Elhoseny, M., Mohammed, M.A., Qabulio, M., & Abdel-Basset, M. (2022). Cost-efficient mobility offloading and task scheduling for microservices IoVT applications in container-based fog cloud network. *ClusterComputin*g, 1-23.

[5] Tuli, S., Gill, S.S., Xu, M., Garraghan, P., Bahsoon, R., Dustdar, S., Sakellariou, R., Rana, O., Buyya, R., Casale, G., & Jennings, N.R. (2022). HUNTER: AI based holistic resource management for sustainable cloud computing. *Journal of Systems andSoftware, 184*, 111124.

[6] Haibeh, L.A., Yagoub, M.C., & Jarray, A. (2022). A survey on mobile edge computing infrastructure: Design, resource management, and optimization approaches. *IEEEAccess, 10*, 27591-27610.

[7] Bentaleb, O., Belloum, A.S., Sebaa, A., & El-Maouhab, A. (2022). Containerization technologies: Taxonomies, applications and challenges. *The Journal ofSupercomputing, 78*(1), 1144-1181.

[8] Chen, X., (2022). Machine learning approach for a circular economy with waste recycling in smart cities. *Energy Reports, 8*, 3127-3140.

[9] Wadhwa, H., & Aron, R. (2022). TRAM: Technique for resource allocation and management in fog computing environment. *The Journal of Supercomputing, 78*(1), 667-690.

[10] Tiwari, R., Mittal, M., Garg, S., & Kumar, S. (2022). Energy-aware resource scheduling in FoG environment for IoT-based applications. *Energy conservation solutions for fog-edge computing paradigms*, 1-19.
[11] Li, Q., Yang, Z., Qin, X., Tao, D., Pan, H., & Huang, Y. (2022). CBFF: A cloud–blockchain fusion framework ensuring data accountability for multi-cloud environments. *Journal of Systems Architecture*, *124*, 102436.
[12] Dehury, C.K., Jakovits, P., Srirama, S.N., Giotis, G., & Garg, G. (2022). TOSCAdata: Modeling data pipeline applications in TOSCA. *Journal of Systems and Software*, *186*, 111164.
[13] Brogi, A., Carrasco, J., Durán, F., Pimentel, E., & Soldani, J. (2022). Self-healing trans-cloud applications. *Computin*g, 1-25.
[14] Karanjai, R., Kasichainula, K., Xu, L., Diallo, N., Chen, L., & Shi, W. (2023). DIaC: Re-Imagining Decentralized Infrastructure As Code using Blockchain. *IEEE Transactions onNetwork and Service Management*.
[15] Toledo, K., Breitgand, D., Lorenz, D., & Keslassy, I. (2023). CloudPilot: Flow acceleration in the cloud. *Computer Networks*, *224*, 109610.
[16] Gupta, M., Bhatt, S., Alshehri, A.H., & Sandhu, R. (2022). Introduction: Requirements for Access Control in IoT and CPS. *In Access Control Models and Architectures For IoTand Cyber Physical Systems* (pp. 1-17). Cham: Springer International Publishing.
[17] Razian, M., Fathian, M., Bahsoon, R., Toosi, A.N., & Buyya, R. (2022). Service composition in dynamic environments: A systematic review and future directions. *Journalof Systems and Software*, *188*, 111290.
[18] Shah, S.D.A., Gregory, M.A., Li, S., dos Reis Fontes, R., & Hou, L. (2022). SDN-based service mobility management in MEC-enabled 5G and beyond vehicular networks. *IEEEInternet of Things Journal*, *9*(15), 13425-13442.
[19] Khan, A.A., Laghari, A.A., Gadekallu, T.R., Shaikh, Z.A., Javed, A.R., Rashid, M., Estrela, V.V., & Mikhaylov, A. (2022). A drone-based data management and optimization using metaheuristic algorithms and blockchain smart contracts in a secure fog environment. *Computers and Electrical Engineering*, *102*, 108234.
[20] Saif, M.A.N., Niranjan, S.K., Murshed, B.A.H., Al-ariki, H.D.E., &Abdulwahab, H.M. (2023). Multi-agent QoS-aware autonomic resource provisioning framework for elastic BPM in containerized multi-cloud environment. *Journal of Ambient Intelligence andHumanized Computing*, *14*(9), 12895-12920.
[21] Li, W., Cao, J., Zhou, B., Deng, S., Zhang, Q., Hu, K., Li, J., &Zhao, H. (2023). Multi-cloud service provision based on decision tree and two-layer Restricted Monte Carlo Tree Search. *Internet of Things*, *22*, 100751
[22] Ullah, A., Kiss, T., Kovács, J., Tusa, F., Deslauriers, J., Dagdeviren, H., Arjun, R., &Hamzeh, H. (2023). Orchestration in the Cloud-to-Things compute continuum: taxonomy, survey and future directions. *Journal of Cloud Computing*, *12*(1), 135.
[23] Kumar, P.H., & AnandhaMala, G.S. (2023). HMAC-R: Hash-based message authentication code and Rijndael-based multilevel security model for data storage in cloud environment. *The Journal of Supercomputing*, *79*(3), 3181-3209.
[24] Wang, X., Guo, P., Li, X., Gangopadhyay, A., Busart, C., Freeman, J., & Wang, J. (2023). Reproducible and Portable Big Data Analytics in the Cloud. *IEEE Transactionson Cloud Computing*.
[25] Ouchaou, L., Nacer, H., & Labba, C. (2022). Towards a distributed saas management system in a multi-cloud environment. *Cluster Computing*, *25*(6), 4051-4071.
[26] Yan, C., & Sheng, S. (2023). Sdn+ K8s Routing Optimization Strategy in 5G Cloud Edge Collaboration Scenario. *IEEE Access*, *11*, 8397-8406.
[27] http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains
[28] Nóvoa, A., Racca, A., &Magri, L. (2024). Inferring unknown unknowns: regularized bias-aware ensemble Kalman filter. *Computer Methods in Applied Mechanics andEngineering*, *418*, 116502.
[29] Sellami, L., &Alaya, B. (2021). SAMNET: Self-adaptative multi-kernel clustering algorithm for urban VANETs. *Vehicular Communications*, *29*, 100332.
[30] Chen, W., &Shi, K. (2021). Multi-scale attention convolutional neural network for time series classification. *Neural Networks*, *136*, 126-140.
[31] Kivi, M.E., &Majidnezhad, V. (2022). A novel swarm intelligence algorithm inspired by the grazing of sheep. *Journal of Ambient Intelligence and Humanized Computing*, *13(*2), 1201-1213.