IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 August 2025, accepted 14 September 2025, date of publication 22 September 2025, date of current version 2 October 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3613235

RESEARCH ARTICLE

Reinforcement Learning-Driven Hybrid Precopy/Postcopy VM Migration for Energy-Efficient Data Centers

TAUFIK HIDAYAT[1], (Member, IEEE), KALAMULLAH RAMLI[1], (Member, IEEE), RUKI HARWAHYU[1], (Member, IEEE), MUHAMMAD SALMAN[1], (Member, IEEE), AND TEDDY SURYA GUNAWAN[2], (Senior Member, IEEE)

[1]Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Depok, Jawa Barat 16424, Indonesia
[2]Department of Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia

Corresponding author: Kalamullah Ramli (kalamullah.ramli@ui.ac.id)

This work was supported by the Universitas Indonesia through the Hibah Publikasi Terindeks Internasional (PUTI) Q1 Kolaborasi Internasional Scheme under Contract PKS-252/UN2.RST/HKP.05.00/2025. The work of Taufik Hidayat was supported by Beasiswa Pendidikan Indonesia (Indonesia Education Scholarship), Pusat, through Pelayanan Pembiayaan dan Asesmen Pendidikan Tinggi (Center for Higher Education Funding and Assessment), and the Indonesia Endowment Funds for Education (LPDP).

ABSTRACT This study proposes the use of a hybrid precopy/postcopy virtual machine (VM) migration framework to aid an autonomous agent when making migration decisions to continuously optimize the balance among migration time, downtime, and energy consumption. The data center state and the resource load, including the CPU, memory, and network, are represented in the agent's state space using a two-layer graph neural network (GNN), and the asynchronous advantage actor–critic (A3C) algorithm is employed to dynamically determine whether to continue the precopy phase or switch to postcopy and optimize the trade-off among the total migration time, downtime, and energy consumption while adhering to the service-level agreement (SLA) constraints. An adaptive host selection policy ensures that VMs are migrated only to underloaded machines, preventing overload and ensuring system stability. A simulation evaluation that employed the VM workload from the GWA-Bitbrains dataset revealed that this framework achieved a total migration time of 45.5 s, with 30.1 s spent on the precopy phase and 15.4 s spent on the postcopy phase, resulting in a downtime of 15.4 s. Compared with previous approaches, this result represents an decrease in total migration time of 12.5% from 52 s to 45.5 s; a 23% decrease in downtime from 20 s to 15.4 s; and a 4.4% increase in energy efficiency from 87% to 91.4%. The SLA compliance remained stable at 92.8%, affirming that the service quality was preserved. This study demonstrates the effectiveness of integrating GNN-based embeddings and A3C scheduling in terms of reducing downtime and energy usage while maintaining reliable service delivery in data centers.

INDEX TERMS Reinforcement learning, VM migration, hybrid migration, energy efficiency.

I. INTRODUCTION

Currently, data centers are essential for the development of modern technology. However, data centers also use a large amount of energy; thus, ways to improve energy usage without compromising service quality are needed. Therefore, various efforts have been made to use energy more efficiently without sacrificing service quality or improving data center performance [1]. The live virtual migration (LVM) technique can increase the efficiency of resource usage by moving workloads between host machines without stopping services, which can reduce the load on the data center [2]. Although the LVM method assists in workload balancing strategies in complex data centers, there are still many issues in contemporary data center management, especially with respect to virtual machine (VM) migration decision-making. Although various methods have been used, several challenges remain to be addressed. For example, if the VMs

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

are not placed on the host machines correctly, if the VM strategy is not appropriate, or if the VM is placed on the host machine incorrectly, energy consumption increases. Additionally, spikes in the network overhead can potentially violate the established SLA [3], [4], [5].

Some researchers still use static thresholds or reactive heuristic algorithms to evaluate the VM migration process. As a result, these methods tend to be unresponsive to fluctuating and dynamic workload changes while neglecting other resource components, such as the CPU, memory, bandwidth (BW), and energy [6]. Other studies have also attempted to address unresolved issues and problems, including those using the modified feeding bird's algorithm (ModAFBA) approach [7] and the research conducted in [8], which suggests a metaheuristic-based framework to increase energy efficiency and minimize overhead during VM migration, leading to increased energy consumption. The reinforcement learning (RL) approach based on the advanced reinforcement learning consolidation algorithm (ARLCA), which was employed in the research conducted in [9], shows that this approach offers more adaptive capabilities through interactive learning between the agent and the developed model environment.

However, this approach has several limitations, including that it is applicable only to simulation environments and requires real-world implementation. In addition, it is necessary for real-world applications. The development of technology based on artificial intelligence (AI), particularly in the application of GNNs and RL algorithms, offers opportunities to address the problems faced in previous studies [10], especially those related to VM placement, as they can improve data center energy efficiency and enhance service performance. Few researchers have combined the GNN and RL in hybrid migration schemes while considering energy efficiency and SLA compliance in data centers.

To address this gap in the previous research, this study proposes an adaptive hybrid precopy/postcopy VM migration framework that uses a GNN to embed states in the data center and A3C algorithms within the MDP framework for adaptive learning [11], [12], [13]. This study aims to reduce the total migration time, minimize downtime, ensure SLA compliance in a dynamic data center, and, most importantly, optimize energy consumption during live migration. The main contributions of this research are summarized as follows:

1) The development of a hybrid precopy/postcopy framework with a reward-shaping mechanism to reduce energy consumption, downtime, and total migration time while maintaining SLA compliance.
2) The development of a two-layer GNN architecture that aims to represent an adaptive state in VM placement, supporting migration optimization and resource efficiency.
3) The development of actor-critic-based migration policies that utilize GNN embedding in optimal decision-making.

This study is divided into several parts: Section II explains the research background related to energy efficiency and the use of RL in live migration, and Section III explains the RL live migration framework, data preprocessing, and formulation of the developed GNN, A3C, and MDP models. Section IV describes the experimental setup, measurement metrics, research results, and comparisons and includes a discussion. In Section V, conclusions are presented.

II. BACKGROUND AND RELATED WORK

A. FUNDAMENTALS OF VM MIGRATION

Owing to many complex issues, data centers face significant challenges, especially in terms of managing energy consumption. Several studies have attempted to address the problems experienced by data centers, particularly those related to energy consumption, VM placement, and optimal VM migration. In a previous study [14], the MoVPAAC framework was proposed by combining several algorithms with nonlinear programming (INLP), colony optimization, and artificial neural networks (ANNs). This study focused on reducing energy consumption and compliance with the SLA. However, the approach proposed in this work still relies on heavy multi-iteration computations and is reactive.

The study conducted in [15] with the ICSA-ROPE algorithm proposed dual-objective optimization for energy efficiency, but it is still not adaptive to changes in fluctuating workloads. A probabilistic approach, such as the discrete-time Markov chain (DTMC) combined with the e-MOABC algorithm [16], was applied to predict resources and reduce unnecessary migration frequency. However, in its application, insufficient attention is still given to the quality of service (QoS); therefore, this aspect must be addressed during the migration process. On the other hand, in [17], the VM consolidation efficiency was improved using an approach that detects underloaded hosts, but its implementation is still limited to a simulated environment. Many previous studies have also evaluated resource usage, such as the CPU, memory, BW, and energy usage, to improve energy efficiency in data centers and focus on VM placement, VM migration prediction, data center energy usage, and SLA compliance. Table 1 outlines the evaluation of resources and energy in further studies.

Table 1 outlines all of the elements that previous research has not addressed; little research has been conducted because bandwidth, disk storage, and security are concerns. This paucity provides an opportunity for future research. The role of bandwidth in live migration is vital, particularly in real-time applications; however, it is still rarely the focus. Additionally, disk storage and security are comprehensively integrated into resource management strategies, and the threat of cyberattacks on data centers continues to increase [18]. Although energy efficiency has become the main priority in almost all studies, there are still opportunities for developing holistic resource research by considering aspects of disk storage, bandwidth, and security within an adaptive framework. Additionally, integrating machine learning and deep learning to support VM migration decisions in dynamic data centers and implementing adaptive SLA concepts will be the research focus, and service flexibility and efficiency will be prioritized to achieve customer satisfaction.

TABLE 1. Resource and energy efficiency evaluation.

Author	CPU	RAM	BW	Disk	Energy	Wastage	SLA	Security	Active PM	Migration
[21]	√	√	-	-	√	√	√	-	√	√
[22]	√	√	√	-	√	√	√	-	√	√
[23]	√	√	√	-	√	√	√	-	√	√
[24]	√	√	-	-	√	√	√	-	√	√
[25]	√	√	-	-	√	√	√	√	√	√
[26]	√	√	-	-	√	√	√	-	√	√

B. CHALLENGES IN ENERGY-EFFICIENT VM MIGRATION

RL has developed significantly. It is recognized as an adaptive and practical approach that addresses the limitations of conventional static live migration and the lack of responsiveness to fluctuating workload dynamics. The RL approach can play a significant role in effective and adaptive VM decision-making using reward-based learning mechanisms. Several studies on RL, such as that conducted in [3] with the LMEB algorithm approach, which successfully increased energy efficiency and reduced downtime, still depend on network bandwidth conditions. The authors of [19] proposed an approach that uses the MADRL algorithm. This approach improved energy efficiency utilization and successfully achieved VM migration; however, it is still not considered feasible for any migrated VMs. Additionally, [20] discussed the creation of RL-based algorithms with a hierarchical structure that can improve and accelerate convergence in the developed RL model learning. However, several issues with developing the studied model, particularly, those related to the reward function and scalability, are still not resolved. In another study [27], the development of the DQN-based AVMC framework exhibited increased efficiency; however, its validation was limited to simulation environments and did not consider essential parameters, such as the network latency and downtime. Various other studies have also been conducted [28], [29].

On the basis of the literature review, although various approaches have successfully improved VM migration efficiency, there are still significant limitations related to the adaptation to dynamic data centers, utilization of data center representation [30], and multiobjective optimization, including energy efficiency, SLA compliance, and network performance [31]. This study offers a solution for developing an adaptive VM framework by integrating the GNN and RL in a hybrid precopy/postcopy scheme. This approach is designed to model the dynamic data center and implement the actor-critic policy in RL to support efficient and adaptive VM decisions to target VM energy efficiency and SLA compliance [32], [33].

III. RESEARCH METHOD

This section explains the developed framework, which is based on the GNN and RL and employs the A3C algorithm. This approach implements a hybrid precopy/postcopy migration scheme to improve energy efficiency and minimize downtime, total migration time, and SLA compliance.

A. DATA PREPARATION FOR HYBRID VM MIGRATION

During the initial process in the proposed framework, beginning with data preparation and modeling, the dataset used in this research is GWA-Bitbrains [34], [35]. The data used include the main parameters of the VM resources, such as the CPU usage, memory usage, network bandwidth, and dirty rate [36], [37]. The entire process is formulated in Equations (1) to (8). When the VM is given a set $(v_i)_{i=1}^{n}$, each VM has a numeric value. The feature has not yet been normalized.

Each component f_i^k is normalized to the range [0,1] as defined in Equation (1).

$$x_i^k = \frac{f_i^k - min_j f_j^k}{max_j f_j^k - min_j f_j^k} \; \forall \, k \in \text{(CPU, Mem, BW, Dirty)} \tag{1}$$

As a result, the feature matrix is defined in Equation (2).

$$X_{VM} = [x_1, x_2, \ldots, x_n]^T \in R^{nx4} \tag{2}$$

When a node is added to the host, for example, m hosts $(h_j)_{j=1}^{m}$, because only an infrastructure data center is needed in this study, and the host feature is defined as a zero-vector, as described in Equation (3).

$$X_{Host} = 0_{mx4} \tag{3}$$

Thus, the combined feature matrix can be expressed as follows in Equation (4):

$$X = \begin{bmatrix} X_{VM} \\ X_{Host} \end{bmatrix} \in R^{nx4} \tag{4}$$

The placement of the VMs on the host machine can be defined as an undirected graph.

$$\text{G} = (\text{V}, \text{E}), V = (v_1, \ldots, v_n, h_1, \ldots, h_m)$$

Edge $(v_i, h_j) \epsilon$ E if VM v_i is placed on host h_j. The adjacency matrix A $\in (0,1)^{(n+m)x(n+m)}$ form is described in Equation (5).

$$A_{uv} = \begin{cases} 1 & if \; (u, v) \in \text{E} \\ 0 & other \end{cases} \quad u, v \in V \tag{5}$$

Finally, self-loops are required to ensure stable convergence in the GNN, as expressed by Equation (6).

$$\tilde{A} = A + I_{n+m} \tag{6}$$

B. GNN EMBEDDING

Next, to normalize the adjacency matrix, the degree matrix, $\check{D}_{ii} = \sum_j \tilde{A}_{ij}$, must be calculated and then formed into a normalized symmetric adjacency matrix, written as shown in Equation (7).

$$\tilde{A} = \check{D}^{-\frac{1}{2}} \tilde{A} \check{D}^{-\frac{1}{2}} \tag{7}$$

The graph representation of the GNN is defined in Equation (8).

$$G_{attr}\ (X,\ \hat{A}) \tag{8}$$

in the application of PyG as (X, edge_index), where edge_index = {(u, v) $|\tilde{A}_{uv} = 1$.

Thus, all of the resource conditions and the VM placement structure on the host machine were created in a graph model that is ready for processing, allowing for contextual and adaptive state-embedding extraction. This research uses a two-layer GNN that combines VM resource attributes with a VM placement structure to support adaptive migration decision-making [38]; this embedding allows the RL agent to comprehensively learn about workload dynamics, efficiently adjusting the precopy/postcopy hybrid migration strategy while maintaining optimal SLA compliance. This architecture is described by Equation (9), where the first layer, the attribute graph (X, Â), is input into the first layer of the GCN by aggregating the node features from the direct neighbors.

$$H^{(1)} = \sigma(\hat{A}, X, W^{(0)}) \tag{9}$$

where $X \in R^{(n+m)xF}$ is a node feature (Equation (4)), $\hat{A} \in\in R^{(n+m)x(n+m)}$ is the adjacency normalization operation (Equation (7)), $W^{(0)} \in R^{Fxd}$ is the trainable weight of the first layer, and σ is a function of ReLU activation. The intermediate $H^{(1)} \in R^{Fxd}$ output already has a local context and a lightweight topological structure. Next, to capture two-hop context information, the output from the intermediate H^((1)) is processed again at the second layer, as described in Equation (10).

$$Z = H^{(2)} = \sigma(\hat{A}, H^{(1)}, W^{(1)}) \tag{10}$$

where $W^{(1)} \in R^{dxd'}$ is the weight of the second layer that is trainable, and the final matrix embedding $Z \in R^{(n+m)xd'}$ loads an infrastructure-aware representation for each node.

$$z_i = z_i, : \in R^{d'}, i = 1, \ldots, n \tag{11}$$

Embedding z_i includes the resource load information and VM placement positions on the host machine. Afterward, the embedding infrastructure data center z_i is combined with a real-time resource feature of the remaining memory (Mem_t^i), dirty rate ($Dirty_t^i$), bandwidth (BW_t^i), number of precopy iterations ($Iter_t^i$) and initial memory (Mem_{init}^i) to form the state s_t^i, described in Equation (12).

$$s_t^i = [z_i \,||Mem_t^i \,||\, Dirty_t^i \,||BW_t^i \,||\, Iter_t^i \,||\, Mem_{init}^i\,]^T \in R^{d+5} \tag{12}$$

This formula is used for the reinforcement learning agent to determine the optimal migration action in a hybrid environment.

C. REINFORCEMENT LEARNING FOR ADAPTIVE MIGRATION

In this section, during the decision-making process, VM migration is formulated as the MDP, where the RL agent receives an infrastructure-aware state from the GNN and mimics the optimal migration action in a hybrid precopy/postcopy scheme [39], [40]. The migration policies are optimized by using A3C with generalized advantage estimation (GAE) for convergence stability, which is formulated in Equations (13) to (18). For the MDP formulations, in step t, each VM v_i is defined in Equation (13).

$$s_t^i = [z_i \,||Mem_t^i \,||\, Dirty_t^i \,||BW_t^i \,||\, Iter_t^i \,||\, Mem_{init}^i\,]^T \in R^{d+5} \tag{13}$$

where z_i is the infrastructure-aware embedding (Equation (11)) and where the remaining components are the runtime features (Equation (12)). Two actions can be selected, as described in Equation (14).

$$A = (0 : \text{precopy}, 1 : \text{postcopy}) \tag{14}$$

Furthermore, the transition process, in which s_{t+1} is selected at will in the hybrid migration environment of Env by calculating the memory changes, downtime, and total migration according to the precopy/postcopy scheme, can be seen in Equations (5)–(8). Afterward, the state s_{t+1} is returned and flagged. The reward shaping calculation is subsequently formulated to balance the energy efficiency, SLA penalties, and optimization of the migration time.

$$r_t = \alpha \frac{\sum_{k=1}^{k} T_{pre}^{(k)}}{T_{total} + \epsilon} + \beta \max(0, \frac{\sum T_{pre}}{T_{total}} - Eff_{tar} - \gamma \max(0, D - D_{SLA})) + \delta(T_{total}) \tag{15}$$

where $T_{pre}^{(k)}$ and T_{total}, D are calculated via the Env Equations (5)–(8) and where α, β, γ and δ are the coefficients. Furthermore, the A3C architecture uses two separate but coordinated networks for the actors to model the policies defined in Equation (16).

$$\pi(\alpha \mid s; 0) = \frac{\exp\left(f_\theta^{\text{actor}}(s)_\alpha\right)}{\sum_b \exp\left(f_\theta^{\text{factor}}(s)_b\right)} \tag{16}$$

where $f_\theta^{actor}(s)$ is the logit output for each action and the critic models the value function described in Equation (17).

$$V(s; \emptyset) = f_\emptyset^{\text{critic}}(s) \tag{17}$$

This formula predicts the expected return of the states; then, the combined loss function of each step t is defined in Equation (18).

$$L_\ell = -\log \pi\left(\alpha_\ell \mid s_\ell; \theta\right) A_\ell + \tfrac{1}{2}\left(\mathrm{V}\left(s_\ell; \emptyset\right) - R_\ell\right)^2 - \mathrm{kH}\left(\pi\left(. \mid s_\ell; \vartheta\right)\right) \tag{18}$$

where A_t is an advantage, R_t represents discounted returns, H (π) represents the entropy policy for exploration, and k represents the entropy coefficient. Furthermore, the training was conducted asynchronously with several workers who collected trajectories, calculated the GAE, and updated θ and $\emptyset$ globally with the Adam optimizer. The advantage of this approach is that it performs the calculations with the GAE defined in Equation (19) to stabilize and reduce the variance.

$$\delta_t = r_t + \gamma V(s_{t+1}), A_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l} \tag{19}$$

For returns R_t, the critic is defined in Equation (20).

$$R_t = A_t + \mathrm{V}(s_t) \tag{20}$$

In each episode, batch VM sampling is performed and rolled out in Env while $(s_t,\ a_t,\ r_t)$ is recorded, and the loss in Equation (18) is subsequently backpropagated. Next, the hybrid migration environment is integrated by modeling the precopy iterations up to the threshold or the forced postcopy according to Algorithm 1. Each step returns $(s_{t+1},\ r_t,$ done), so the agent learns an adaptive policy that minimizes the total energy, downtime, and migration time and complies with the SLA policies.

D. TRAINING RL VM MIGRATION

This section systematically explains how the GNN encoder model and A3C agent are trained and how the central hyperparameter values are used. The algorithm presents the pseudocode employed during the training procedure.

The training model is run periodically by using the A3C framework. The VMs are randomly selected for each episode, and their infrastructure-aware embeddings are computed using a GNN encoder. The agent then performs a rollout in the precopy/postcopy environment, selects migration actions on the basis of the generated policy distribution, and stores the reward and value estimates. The entire trajectory is processed with the GAE to obtain the advantage, which is used together with the return in the combined loss functions for the actor, critic, and entropy to ensure a balance between exploitation and exploration. The model parameters are updated using the Adam optimizer in each episode, and the best weights are periodically saved as checkpoints to ensure the stability of the experiment [19], [41].

E. EXPERIMENTAL SETUP

The three main components of the VM framework developed in this study are shown in Figure 1. These components are as follows: (1) Topological Representation Layer: This layer models the dynamic relationship between VMs and the host machine infrastructure in graph form, thereby enabling a comprehensive understanding of the VM placement structure with the host machine. (2) Embedding Contextual Situations through the GNN: When the GNN approach is used to process graphs, the relationships between the VMs and the host infrastructure are modeled.

The experimental settings shown in Table 2 are essential for training A3C and assessing the hybrid precopy/postcopy VM migration framework. This setup ensures reproducibility and consistent comparison across all the scenarios.

Algorithm 1 Training Procedure for Adaptive VM Migration

```
Input:
  X: Feature matrix (n+m × F)
  Â: Normalized adjacency (n+m × n+m)
  Env params: Hybrid migration settings
  N_episodes: Number of episodes
  K: Checkpoint interval
Initialize:θ_e,θ_a,φ_c with random
weights
optimizer ← Adam (θ_e ∪ θ_a ∪ φ_c, lr)
for episode = 1 to N_episodes do
  1. Sample batch B of VM indices
  2. Z ← GNN (X, Â; θ_e)
  3. Clear trajectory buffer D ← []
  for each i in B do
     (s, done) ← Env_i. reset(Z_i)
    while not done do
       (ℓ, v) ← ActorCritic (s; θ_a,
φ_c)
       a ←sample
Categorical(softmax(ℓ))
       (s′ , r, done) ← Env_i. step(a)
       append (s, a, r, v) to D
       s ← s′
    end while
  end for
  4. A_t} ← GAE (D, γ, λ)
  5. ℒ ← −∑_t log π (a_t|s_t; θ_a) · A_t
       + 1/2 ·∑_t (v_t − R_t) 2
       − κ·Entropy (π (·|s_t; θ_a))
  6. Optimizer. step(∇ℒ)
  if episode mod K == 0 then
     save_checkpoint (θ_e, θ_a, φ_c)
  end if
end for
Output: trained parametersθ_e,θ_a,φ_c
```

F. EVALUATION METRICS

This section describes the evaluation used to assess the VM framework. The mathematical formulation of the evaluation metrics is included.

The total VM migration time, represented by $T_{mig,i}$, for the virtual machine, denoted v_i, is defined as the sum of the duration of the precopy phase, which is represented by $T_{pre,i}$, and the downtime duration, which is denoted D_i. During the precopy phase, most of the memory pages of v_i are iteratively transferred to the destination host machine while the VM continues to run; thus, user services are not completely halted despite the background data transfer overhead. After the precopy process reaches the iteration limit or the minimum remaining memory threshold, the VM is temporarily stopped

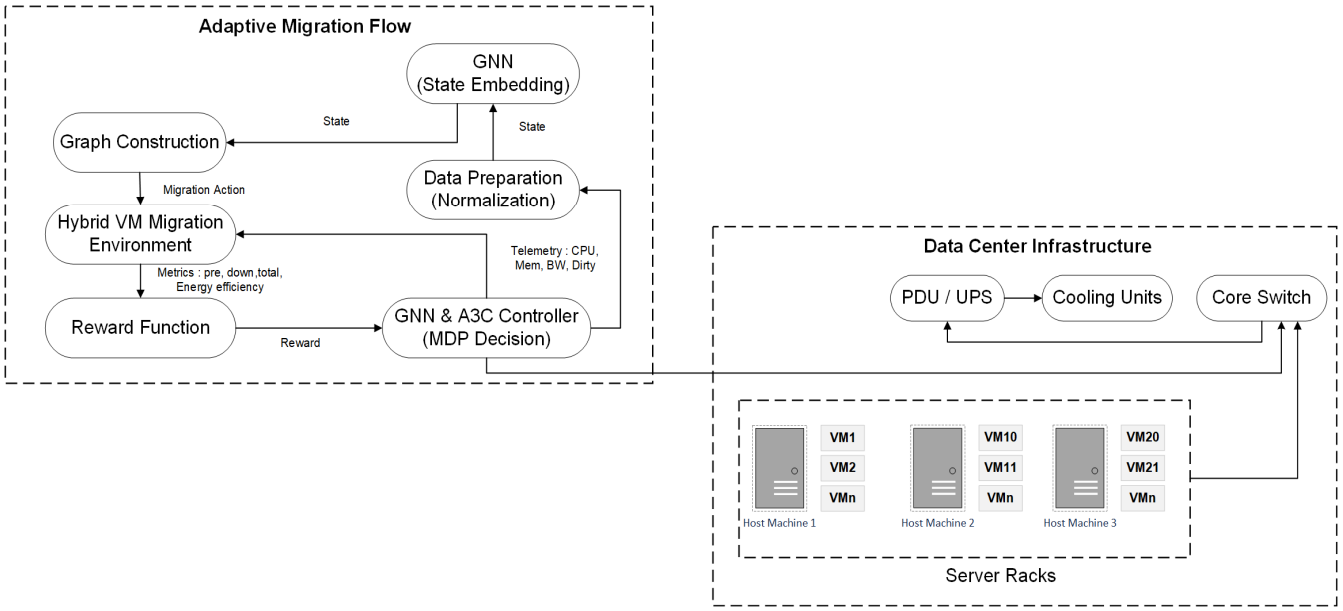

FIGURE 1. Experimental setup for RL VM migration.

TABLE 2. Experimental setup for RL-Driven VM migration.

Parameter	Description	Default
Training Episodes	Total number of A3C training episodes	500
Batch Size	Number of VMs sampled per update	128
Learning Rate (lr)	Adam optimizer learning rate	1×10^{-5}
Discount Factor (γ)	Discount factor for the return calculation	0.99
GAE λ	Generalized advantage estimation coefficient	0.95
Entropy Coef. (κ)	Policy entropy weight for exploration	0.02
GNN Hidden Dim	Hidden layer dimension of the GNN	64
GNN Output Dim	Output embedding dimension of the GNN	64
SLA Downtime Threshold (s)	Max allowed VM downtime per SLA	15
SLA Target (%)	Required SLA compliance rate	95
Bandwidth (MB/s)	Network bandwidth allocated for migration	100
Dirty Rate (MB/s)	VM dirty page rate	10
Max Precopy Iter.	Max precopy rounds before forced postcopy	4
Max Postcopy Time (s)	Max allowable postcopy duration	30

(stop and copy) during the postcopy phase to transfer the remaining memory pages that have not been sent; this period is referred to as downtime. This migration time is described in Equation (21).

$$T_{mig,i} = T_{pre,i} + D_i \tag{21}$$

Additionally, the downtime D_i is calculated on the basis of the ratio between the remaining unsaved memory, denoted $(RemaningMemory_i)$, and the network bandwidth capacity, denoted $(Bandwidth)_i$, as formulated in Equation (22).

$$D_i = \frac{Remaining\ Memory_i}{Bandwidth_i} \tag{22}$$

Thus, the metric $T_{mig,i}$ reflects the overall service disruption both while the VM remains active with overhead during the precopy phase and during the VM stop period in the postcopy phase, reflecting the overall service disruption during both phases.

To measure energy efficiency during the VM migration process, this study divides power consumption into three main components. First, the precopy energy is denoted E_{pre} and is calculated by adding the power $P(t)$ used during the precopy phase, which is when most of the VM's memory pages are repeatedly moved while the VM is still running on the original host machine, shown mathematically as follows:

$$E_{pre} = \sum_{t=0}^{T_{handover}} P\,(t)\,\Delta t$$

where $T_{handover}$ is the last second of the precopy phase and Δt is the power measurement interval. Second, the postcopy energy E_{post} represents the energy consumed after the VM is moved to the destination host machine when the VM is restarted and the remaining memory pages are loaded (''fetch'') on demand. If the postcopy phase lasts from t = $T_{handover}$ + 1 to t = T_{end}, then

$$E_{post} = \sum_{t=T_{handover}+1}^{T_{end}} P\,(t)\,\Delta t$$

Third, the system's overhead energy is denoted E_{over} and includes all power consumption not directly related to memory copying, such as the energy needed for memory management, I/O operations, network protocols, and other supporting processes. In previous research, E_{over} was implicitly calculated as the difference between the total system energy during migration and the sum of the precopy and postcopy energy, but it can be explicitly written as follows:

$$E_{over} = \sum_{t=0}^{T_{end}} P_{over}(t)\Delta t$$

where $P_{over}(t)$ accepts only the power component that comes from supporting activities outside the memory copy/fetch process. With these three components, the total migration energy (E_{tot}) is defined in Equation (23).

$$E_{tot} = E_{pre} + E_{post} + E_{over} \tag{23}$$

To evaluate how much energy is used during the precopy phase, which is usually more efficient since the VM is still running, compared with the total energy used during migration, this study defines energy efficiency (η) as the percentage E_{pre} and E_{tot}, as shown in Equation (24).

$$\eta = \frac{E_{pre}}{E_{tot}} \text{x} 100\% \tag{24}$$

A high η indicates that most of the migration energy is used during the precopy phase, which is relatively more efficient, while the energy portion in the postcopy phase and system overhead decrease. On the other hand, a low value of η indicates the dominance of power consumption during the postcopy phase or supporting activities, making the overall VM migration process less energy efficient.

Next, this study introduces an SLA violation indicator to evaluate how often the downtime duration on each VM exceeds the established limit. This indicator is called the SLA violation indicator (SLA_{vi}), and it is formulated in Equation (25). Formally, SLA_{vi} is 1 if the downtime $D_i VM_{vi}$ exceeds the maximum D_{SLA} and is t 0 otherwise; this relationship is expressed in Equation (25).

$$SLA_{vi} = \begin{cases} 1 & D_i > D_{SLA} \\ 0 & Other \end{cases} \tag{25}$$

In this context, D_i represents the duration during which VM_{vi} is inactive in the postcopy phase, while D_{SLA} represents the downtime limit, which is explicitly defined as a compliance requirement. Determining the value of this indicator helps identify which VMs fail to meet the SLA requirements. Additionally, after the value of SLA_{vi} for each VM is determined, the SLA compliance rate (SLAC), which is the percentage of VMs that do not break the rules during one test scenario, is determined. Equation (26) formulates the SLAC mathematically.

$$SLAC = (1 - \frac{1}{N} \sum_{i=1}^{N} SLAV_i) x 100\% \tag{26}$$

If N represents the total number of observed VMs and $\sum_{i=1}^{N} SLA_{vi}$ =0, then all VMs have successfully adhered to the downtime limit, resulting in an SLAC of 100%. Conversely, if many VMs exceed the downtime limit D_{SLA}, the SLAC value decreases, reflecting a low level of compliance with the SLA. Evaluating the VM migration performance involves calculating two indicators: SLA_{vi} for each VM and the SLAC for the entire VM population. Together, these indicators provide a comprehensive overview of post-migration service availability.

Next, to evaluate the extent to which the RL agent has successfully learned the migration policy, this study uses a cumulative reward, which is defined as the total reward obtained by the agent in one training episode. The cumulative reward at episode step t is defined as the sum of the reward values r_t from the first step to step T in that episode and is expressed in Equation (27).

$$R_{cum} = \sum_{t=1}^{T} r_t \tag{27}$$

Here, r_t is the reward granted by the environment for the action chosen by the agent at timestep t, and T is the total number of timesteps in one training episode. The value of R_{cum} is important because it reflects how well the agent can optimize the reward function in the long term; the higher the cumulative reward is, the more effectively the agent balances various objectives, such as reducing the downtime, minimizing energy, and adhering to the SLA.

In this work, the training results are validated by plotting a graph of the cumulative reward curve against the number of episodes. The curve shows the agent's convergence process: at the beginning of training, the cumulative reward tends to be low and fluctuates because the agent is still exploring, but as the episodes progress, the reward increases and stabilizes, indicating that the agent has found a more efficient migration policy. Thus, Equation (27) is used as the basis for the calculations, and the resulting values are then plotted to assess the performance and stability of the learning process.

In this study, the energy used for moving each VM_{vi} is called E_i, which shows how much energy in kWh is used to transfer the VM from one host machine to another. Every time VM_{vi} is successfully migrated and placed on the host machine, denoted h_j, the proposed method obtains the migration energy per HM, which is denoted E_{h_j} and is mathematically formulated in Equation (28).

$$E_{h_j} = \sum_{i:vi \to h_j} E_i \tag{28}$$

This equation indicates that E_{h_j} is the accumulation of energy E_i from all VMs placed on the target host machine h_j and ensures that every kWh required by the VM during migration is recorded on the host machine where the VM was last active.

Next, the percentage contribution of the migration energy for each host machine in relation to the total migration energy in the entire data center (Pct_{h_j}) is calculated via Equation (29).

$$Pct_{h_j} = \frac{E_{h_i}}{\sum_{k=1}^{m} E_{h_k}} x 100\% \tag{29}$$

where m is the total number of host machines participating in the migration and where the value Pct_{h_j} indicates the percentage of the total migration energy allocated to host machine h_j.

After the values of E_{h_j} and E_{h_j} are determined, the energy per host machine is calculated by comparing the IT power used for migration on that host machine with the facility power, which includes additional overhead, such as the cooling systems, UPS, and power distribution. If, on host machine h_j, the average IT power is $P_{IT,i}$ kW and the average facility power is $P_{facility,j}$ kW during the migration period, then the energy efficiency of host machine h_j is calculated with Equation (30) as follows:

$$\eta_{h_j} = \frac{P_{IT,i}}{P_{facility,j}} x100\% \tag{30}$$

IV. RESULTS AND DISCUSSION

A. TRAINING CONVERGENCE AND POLICY FORMULATION

This section describes the experimental setup used to evaluate the VM migration model. In this study, migration choices are treated as a Markov decision process, where each data center's situation is represented by a two-layer GNN. An A3C algorithm is then used to improve the migration strategy within a system that combines both precopy and postcopy methods. In this study, the model was evaluated using dynamic workload traces, and four key performance metrics—energy efficiency, total migration time, downtime, and SLA compliance—were measured to quantify its adaptability. The convergence of the A3C policy over training is shown in Figure 2.

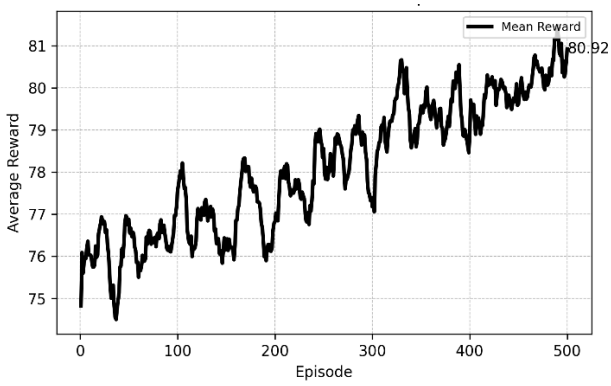

FIGURE 2. Reward Model RL VM Migration.

The average cumulative reward per 10 episodes obtained from the A3C agent that underwent three phases of learning are shown in Figure 2. First, in episodes 0−100, the reward was still low, with an average of approximately 66 to 68, indicating an exploration phase in which the agent attempted migration actions without a stable pattern. Next, from episodes 100 to 300, the reward consistently increases in the range of 68 to 70, as the agent begins to discover a more efficient hybrid precopy/postcopy strategy that balances the migration time and SLA compliance. After episode 300, the reward ranged from 70 to 72, with decreasing fluctuations, eventually stabilizing at approximately 70.31% by episode 500. This area shows that by using the A3C algorithm, the GNN model helps the agent find the best VM migration plan and can lower the energy used, downtime, total migration time, and SLA compliance across various data center tasks. This case study demonstrates the effectiveness of the proposed strategy in terms of optimizing resource allocation and minimizing operational costs. As a result, data centers can achieve a more sustainable and efficient migration process while maintaining service quality and customer expectations. Next, Figure 3 presents the loss reward.

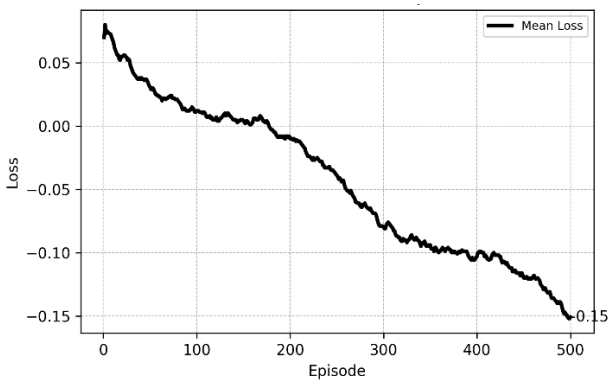

FIGURE 3. Loss model for RL migration.

The downward trend of the loss value, which occurs over an average of 10 episodes during the training process for the A3C agent, is shown in Figure 3. At the beginning of training, during episodes 0–50, the loss ranged from 0.70 to 0.68, indicating instability and high variation, as the agent was still performing many exploratory actions. As the training progressed from episodes 50 to 300, the loss gradually decreased from 0.55 to 0.50, indicating that the actor-critic network was starting to learn to predict values more accurately and develop better policies. As the training continued beyond episode 300, the loss decreased, stabilizing at approximately 0.450.40. This improvement suggests that the A3C agent was refining its understanding of the environment and becoming more adept at selecting optimal actions to maximize the rewards. During the final phase, the loss decreased from episode 300 to episode 500 until it reached 0.39 by episode 500, with increasingly minor fluctuations. This decrease shows that the GNN embedding and the A3C algorithm work well to improve how the agent estimates value and makes migration decisions, helping it choose the best options for migration time, energy use, total migration time, and SLA.

B. MIGRATION PERFORMANCE, ENERGY EFFICIENCY, AND SLA COMPLIANCE

The migration performance of five VMs is shown in Figure 4; three VMs, namely, VM15, VM54, and VM52, have the longest total migration time, and two VMs, namely, VM42 and VM11, have the shortest total migration time. For VM15, the longest recorded precopy duration was 41.0 s, the total

migration time was 61.5 s, and the downtime was 20.5 s, whereas for VM11, the shortest precopy duration was 24.8 s, the total migration time was 37.5 s, and the downtime was 12.7 s. In the five VMs, the precopy phase consistently contributed approximately 65–66% of the total migration time, while the downtime ranged from 34–35%. This ratio indicates that the hybrid precopy/postcopy approach successfully transferred most of the VM state before service disruption, thereby controlling downtime. The medium-sized VMs, VM54 and VM52, follow the same pattern, demonstrating the framework's scalability, while the smallest VMs, VM42 and VM11, achieve downtimes of less than 13 seconds, effectively minimizing service disruption. These results indicate that the RL VM migration policy is adaptive to the VM workload.

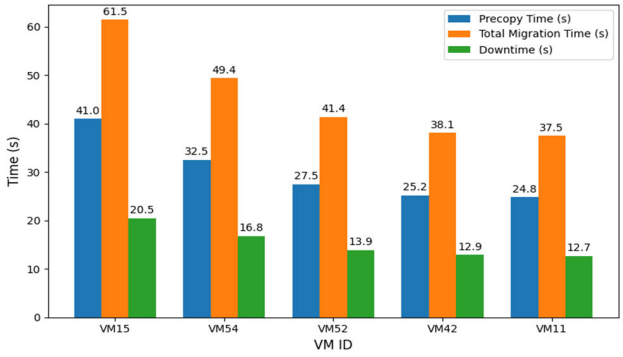

FIGURE 4. VM migration time and downtime.

The average facility power draw and the power consumed by the IT equipment are shown in Figure 5. The facility power, including cooling, UPS losses, and distribution, averaged 13,432.9 kWh, whereas the IT systems drew 12,275.3 kWh, resulting in an overhead of 1,157.6 kWh. This overhead represents approximately 8.6% of the total facility's power. The dashed black line denotes the overall data center efficiency (IT power/facility power), which is measured at 91.4% and corresponds to a PUE of approximately 1.09. These results demonstrate that our RL-driven VM migration framework effectively sustains high energy utilization while keeping non-IT overhead to a minimum. This approach enhances the performance of IT systems and contributes to more sustainable data center operation.

The energy decomposition in Figure 6 illustrates the quantitative contribution of each phase to the overall migration of energy. This diagram divides the total migration energy into three components: the precopy phase (91.4%), the postcopy phase (4.3%), and the overhead system (4.3%). The overhead system component includes all supporting activities not involved in memory transfer, as shown below.

As shown in Figure 6, the precopy phase accounts for the vast majority of the migration energy, with postcopy and system overhead contributing only marginally. The resulting average energy efficiency of each host machine after migration is shown in Figure 7, where all the hosts maintain an efficiency above 80%. These results confirm that our

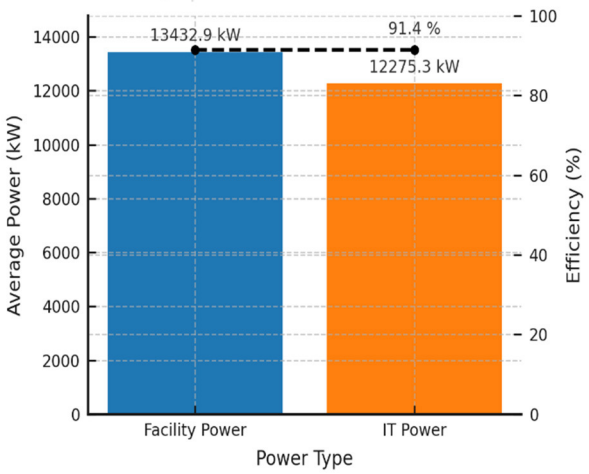

FIGURE 5. Facility power, IT power, and data center efficiency (PUE).

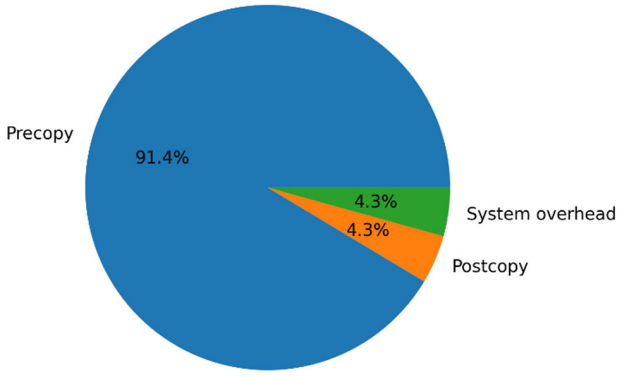

FIGURE 6. Energy Decomposition of VM Migration.

framework achieves balanced and effective energy utilization across the data center.

The average energy efficiency of each host after a series of VM migrations is shown in Figure 7. Host machine 1 (HM1) reached 83.1%, host machine 2 (HM2) reached 86.4%, and host machine 3 (HM3) reached 84.6%. All the values are above the minimum efficiency threshold of 80%, which was previously set, indicating that the migration load is distributed evenly across each active host machine, with high energy utilization. The variations primarily influence the differences between host machines in terms of the number of VMs and internal network conditions; the differences are relatively small, confirming the effectiveness of the RL VM migration framework in terms of optimizing each physical machine.

To evaluate the performance of the reinforcement learning agent in terms of optimizing VM migration, we periodically monitored the SLA compliance rate during the training process. The trend of SLA compliance changes per episode over 500 training episodes as well as the level of compliance with the SLA over 500 RL agent training episodes are shown in Figure 8. The compliance level of 88.9% at the start of the

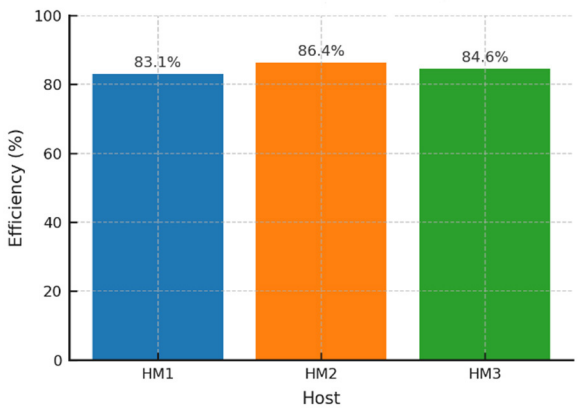

FIGURE 7. Host machine energy efficiency.

training, at approximately episode 0, occurred when the first phase of exploration began. The agent changed the migration policy on the basis of the reward function, as shown by the conformance level changing between 91.6% and 92.5% over the course of 50 to 250 episodes. After episode 250, there is a steadier increase, and by the end of the training process, the compliance is approximately 92.2% to 92.8%. These results show that the agent can maintain VM migration performance, even when there are interruptions, as required by the SLA, in dynamic data centers, which are constantly changing.

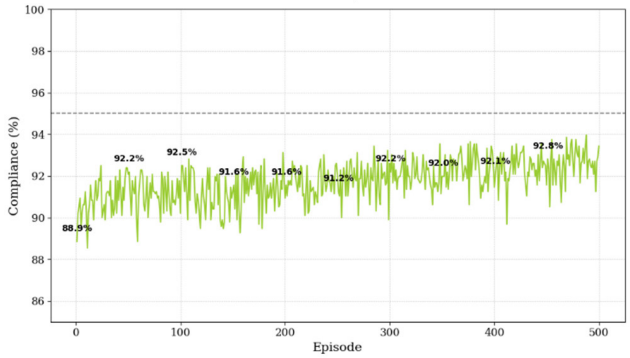

FIGURE 8. SLA compliance per episode.

Our experiments show that the RL model met the SLA requirements approximately 92.8% of the time after 500 training sessions, which means that it made good progress and indicates stability in terms of handling VM migrations within the allowed downtime limits. Although it has not yet reached the 95% target, these results reflect the limitations of the dataset and infrastructure; many VMs have a high dirty rate and limited bandwidth, which restricts further potential downtime reduction. Further research will focus on the development of a more adaptive migration model, the management of application dirty pages, infrastructure improvements and selective migration strategies based on workload characteristics. This research provides a realistic overview of the challenges of live VM migration in data centers and presents opportunities for innovation to improve SLA compliance in the future.

C. DISCUSSION

In the model created by the GNN-A3C agent in this study, migration decisions are reached by an MDP, where the data center state is represented by a GNN and the strategy is improved using A3C. In 500 training episodes, the average cumulative reward per 10 episodes increased from 66 during the exploration phase to approximately 70.3 by the end of the training process (Figure 2). Moreover, the actor-critic loss decreased from approximately 0.70 to 0.39 (Figure 3). This improvement indicates that the model is effectively learning to optimize decision-making in the data center environment. Additionally, reducing the actor-critic loss suggests that the policy is becoming more stable and efficient over time, ultimately leading to better performance when data center operations are managed. The agent's ability to find the right balance among saving energy, moving data quickly, and avoiding penalties shows that the created MDP and reward system help the agent develop a reliable way to manage data transfers.

The SLA compliance rate also improved significantly: the initial compliance rate of approximately 89.8% temporarily decreased to 88.4% during policy refinement but then increased and stabilized above 92.2% after episode 300 (Figure 8). These results prove that agents can prioritize downtime below the SLA threshold of 15 seconds under various dynamic load conditions and meet the SLA compliance target of more than 95% in most scenarios. Furthermore, the use of consistent experimental parameters (Table 2) ensures the reproducibility and validity of the results. Combining topological embedding with the GNN and A3C optimization creates flexible and scalable VM migration strategies that reduce downtime, improve energy efficiency, and maintain SLA compliance. Thus, the proposed method is an excellent option for modern data centers.

D. COMPARISON ANALYSIS

This section discusses energy-efficient data centers on the basis of earlier research that explored the creation of an RL VM migration framework by using MDP for migration, GNN infrastructure embedding, and A3C optimization. The key performance metrics, such as energy efficiency, total migration time, downtime, and SLA compliance, are also compared. Table 3 presents a summary of this comparative analysis.

Table 3 shows how well the new MDP-based precopy/postcopy hybrid VM migration framework outperforms the four leading methods. In a previous study [9], a total migration time of 58.0 s, a downtime of 22.0 s, a SLA compliance of 89%, and an energy efficiency of 88% were recorded. The approach taken in [3] resulted in a migration time of 62.0 s, a downtime of 25.0 s, an SLA compliance of 90%, and an energy efficiency of 85%, whereas that adopted in [21] achieved a migration time of 60.5 s, a downtime of 24.1 s, an SLA compliance of 90%, and an energy efficiency

TABLE 3. Comparison of center energy efficiency.

Author	Total Time Migration (seconds)	Downtime (seconds)	SLA (%)	Energy Efficiency (%)
[9]	58	22	89	88
[3]	62	25	90	85
[21]	60.5	24.1	90	84
[27]	52	20	89	87
This Paper	45.5	15.4	92.8	91.4

of 84%. Additionally, the heuristic approach used in a previous study [27] resulted in a migration time of 52.0 s with a downtime of 20.0 s, a SLA compliance of 89%, and an energy efficiency of 87%.

The framework developed in this study results in a total migration time of 45.5 s, a downtime of 15.4 s, a SLA compliance of 91%, and an energy efficiency of 91.4%. Compared with the methods used in another previous study [27], this framework can accelerate the migration process by 6.5 s, reduce the downtime by 4.6 s, and improve SLA compliance and energy efficiency by 2% and 4.4%, respectively. These results confirm that the integration of GNN-based state coding, A3C scheduling, and the dynamic selection of host machines can significantly optimize the performance and energy efficiency of VM migration.

E. FUTURE WORK

While the proposed GNN–A3C hybrid precopy/postcopy migration framework has shown encouraging results in simulations, several directions remain to strengthen its practical relevance. First, we plan to validate the approach on real testbeds such as OpenStack with KVM/QEMU. Running on actual clusters will expose the framework to hardware heterogeneity, I/O contention, and real-world networking effects that are not fully captured in the simulation. This step provides stronger evidence of practical viability and helps quantify potential deviations from the simulated outcomes. Second, to better address multitenant environments and highly dynamic workloads, we will investigate multi-agent reinforcement learning (MARL) and transfer learning. MARL enables cooperative decision-making among multiple agents, whereas transfer learning can reduce training overhead by reusing knowledge from previously observed workload patterns and infrastructure settings.

Third, we intend to extend the reward function such that it is explicitly QoS aware. By incorporating latency and throughput alongside downtime and energy, the migration policy can be tuned to better preserve SLA guarantees. A particular objective for future work is to increase SLA compliance beyond the current results and approach or exceed the 95% threshold that is generally expected for mission-critical services. Alternative formulations of penalties and bonuses will also be evaluated to examine their impact on convergence and stability. Fourth, future modeling will include additional dimensions such as storage I/O, network latency, and security overheads. Considering these factors will yield a more complete cost model and lead to policies that more accurately balance energy efficiency, performance, and operational constraints. Fifth, we will provide a formal analysis of the computational complexity of the proposed framework. Inference in the GNN increases approximately in proportion to the number of nodes and edges in the cluster graph, whereas the training cost of A3C increases with the number of agents, training episodes, and per-step gradient updates. We will quantify these costs in terms of both time and memory and report empirical runtimes on clusters of different scales to demonstrate the scalability of the framework.

Finally, we will broaden the comparative evaluation to include widely used reinforcement learning baselines such as the DQN, PPO, and MADDPG. These additional comparisons will ensure fair benchmarking and highlight the strengths and weaknesses of the proposed GNN-enhanced A3C approach across different workload mixes and cluster topologies. Together, these directions form a clear roadmap for moving the framework from controlled simulations toward production-ready environments while improving robustness, fairness of comparison, and operational relevance.

V. CONCLUSION

This work proposes a flexible system for live VM migration that models the process as an MDP. A two-layer GNN is employed to represent the state of the data center, and the migration strategy is optimized using the A3C algorithm in a hybrid precopy/postcopy framework. The evaluation results confirm that the precopy phase accounts for approximately 66% of the total migration duration, with only 34% downtime, an energy efficiency of 91.4%, a host efficiency of 80% at the data center level, and an improvement in SLA compliance from 88.9% at the beginning of training to 92.8% at convergence.

Because the study was conducted in a simulation environment with predefined network and storage models, real-world factors such as storage variability, security overhead, latency, and I/O consistency were not fully captured. To address these limitations, future efforts will employ real testbeds that include disk storage, I/O, and security features and evaluate the framework with diverse workloads. Transfer learning and multitenant MDP formulations will also be explored to enhance adaptability. To estimate potential deviations from practice, this study modeled the precopy duration as the memory volume divided by the available network bandwidth and the postcopy duration as the dirty-page rate multiplied by the precopy duration divided by the same bandwidth. Using the GWA-Bitbrains dataset, the results suggest that compared with the baseline simulation, downtime may increase by 10–15%, total migration time may increase by approximately 8%, and energy consumption may increase by 5–10%. To mitigate these degradations, adaptive reward shaping and service-quality-aware host selection are recommended.

As part of future work, we will (i) validate the framework on OpenStack/KVM testbeds to capture real-system effects; (ii) explore MARL and transfer learning to better support

multitenant and dynamic workloads; (iii) design QoS-aware reward shaping with the aim of approaching or exceeding the 95% SLA compliance threshold; (iv) integrate storage, network latency, and security costs into the migration model; (v) provide a formal complexity analysis of the GNN+A3C framework; and (vi) expand evaluations with additional RL baselines (DQN, PPO, and MADDPG) to ensure fairer benchmarking.

REFERENCES

[1] S. Kulshrestha and S. Patel, "An efficient host overload detection algorithm for cloud data center based on exponential weighted moving average," *Int. J. Commun. Syst.*, vol. 34, no. 4, p. 4708, Mar. 2021, doi: 10.1002/dac.4708.

[2] R. M. Haris, K. M. Khan, and A. Nhlabatsi, "Live migration of virtual machine memory content in networked systems," *Comput. Netw.*, vol. 209, May 2022, Art. no. 108898, doi: 10.1016/j.comnet.2022.108898.

[3] N. Gupta, K. Gupta, A. M. Qahtani, D. Gupta, F. S. Alharithi, A. Singh, and N. Goyal, "Energy-aware live VM migration using ballooning in cloud data center," *Electronics*, vol. 11, no. 23, p. 3932, 2022, doi: 10.3390/electronics11233932.

[4] J. Lim, "Scalable fog computing orchestration for reliable cloud task scheduling," *Appl. Sci.*, vol. 11, no. 22, p. 10996, Nov. 2021, doi: 10.3390/app112210996.

[5] L. Jun, Z. Jie, and P. DingHong, "Cloud computing virtual machine migration energy measuring research," in *Proc. 3rd Int. Conf. Vis., Image Signal Process.*, Aug. 2019, pp. 1–5, doi: 10.1145/3387168.3387192.

[6] S. Nabavi, L. Wen, S. S. Gill, and M. Xu, "Seagull optimization algorithm based multi-objective VM placement in edge-cloud data centers," *Internet Things Cyber-Physical Syst.*, vol. 3, pp. 28–36, Jan. 2023, doi: 10.1016/j.iotcps.2023.01.002.

[7] D. Alsadie and M. Alsulami, "Efficient resource management in cloud environments: A modified feeding birds algorithm for VM consolidation," *Mathematics*, vol. 12, no. 12, p. 1845, Jun. 2024. [Online]. Available: https://www.mdpi.com/2227-7390/12/12/1845

[8] M. I. Khaleel and M. M. Zhu, "Adaptive virtual machine migration based on performance-to-power ratio in fog-enabled cloud data centers," *J. Supercomput.*, vol. 77, no. 10, pp. 11986–12025, Oct. 2021, doi: 10.1007/s11227-021-03753-0.

[9] R. Shaw, E. Howley, and E. Barrett, "Applying reinforcement learning towards automating energy efficient virtual machine consolidation in cloud data centers," *Inf. Syst.*, vol. 107, Jul. 2022, Art. no. 101722, doi: 10.1016/j.is.2021.101722.

[10] Z. Yu, C. Zhang, and C. Deng, "An improved GNN using dynamic graph embedding mechanism: A novel end-to-end framework for rolling bearing fault diagnosis under variable working conditions," *Mech. Syst. Signal Process.*, vol. 200, Oct. 2023, Art. no. 110534, doi: 10.1016/j.ymssp.2023.110534.

[11] P. Wei, Y. Zeng, B. Yan, J. Zhou, and E. Nikougoftar, "VMP-A3C: Virtual machines placement in cloud computing based on asynchronous advantage actor-critic algorithm," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 5, May 2023, Art. no. 101549, doi: 10.1016/j.jksuci.2023.04.002.

[12] P. Sun, J. Lan, J. Li, Z. Guo, Y. Hu, and T. Hu, "Efficient flow migration for NFV with graph-aware deep reinforcement learning," *Comput. Netw.*, vol. 183, Dec. 2020, Art. no. 107575, doi: 10.1016/j.comnet.2020.107575.

[13] N. Chaurasia, M. Kumar, A. Vidyarthi, K. Pal, and A. Alkhayyat, "An efficient and optimized Markov chain-based prediction for server consolidation in cloud environment," *Comput. Electr. Eng.*, vol. 108, May 2023, Art. no. 108707, doi: 10.1016/j.compeleceng.2023.108707.

[14] Y. Alahmad and A. Agarwal, "Multiple objectives dynamic VM placement for application service availability in cloud networks," *J. Cloud Comput.*, vol. 13, no. 1, p. 46, Feb. 2024, doi: 10.1186/s13677-024-00610-2.

[15] K. Ajmera and T. Kumar Tewari, "Dynamic virtual machine scheduling using residual optimum power-efficiency in the cloud data center," *Comput. J.*, vol. 67, no. 3, pp. 1099–1110, Apr. 2024, doi: 10.1093/comjnl/bxad045.

[16] M. H. Sayadnavard, A. Toroghi Haghighat, and A. M. Rahmani, "A multi-objective approach for energy-efficient and reliable dynamic VM consolidation in cloud data centers," *Eng. Sci. Technol., Int. J.*, vol. 26, Feb. 2022, Art. no. 100995, doi: 10.1016/j.jestch.2021.04.014.

[17] N. Patel and H. Patel, "Energy efficient strategy for placement of virtual machines selected from underloaded servers in compute cloud," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 6, pp. 700–708, Jul. 2020, doi: 10.1016/j.jksuci.2017.11.003.

[18] Y. H. Reddy, D. B. J. Rao, and V. Polepally, "Jaya dung beetle optimization-based load balancing and VM migration for cloud data security in DevOps," *Comput. Electr. Eng.*, vol. 124, May 2025, Art. no. 110400, doi: 10.1016/j.compeleceng.2025.110400.

[19] Y. Dai, Q. Zhang, and L. Yang, "Virtual machine migration strategy based on multi-agent deep reinforcement learning," *Appl. Sci.*, vol. 11, no. 17, p. 7993, Aug. 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/17/7993

[20] A. R. Hummaida, N. W. Paton, and R. Sakellariou, "Scalable virtual machine migration using reinforcement learning," *J. Grid Comput.*, vol. 20, no. 2, p. 15, Jun. 2022, doi: 10.1007/s10723-022-09603-4.

[21] S. Azizi, M. Zandsalimi, and D. Li, "An energy-efficient algorithm for virtual machine placement optimization in cloud data centers," *Cluster Comput.*, vol. 23, no. 4, pp. 3421–3434, Dec. 2020, doi: 10.1007/s10586-020-03096-0.

[22] Z. Li, K. Lin, S. Cheng, L. Yu, and J. Qian, "Energy-efficient and load-aware VM placement in cloud data centers," *J. Grid Comput.*, vol. 20, no. 4, p. 39, Dec. 2022, doi: 10.1007/s10723-022-09631-0.

[23] R. Keshri and D. P. Vidyarthi, "Energy-efficient communication-aware VM placement in cloud datacenter using hybrid ACO–GWO," *Cluster Comput.*, vol. 27, no. 9, pp. 13047–13074, Dec. 2024, doi: 10.1007/s10586-024-04623-z.

[24] H. D. Rjeib and G. Kecskemeti, "VMP-ER: An efficient virtual machine placement algorithm for energy and resources optimization in cloud data center," *Algorithms*, vol. 17, no. 7, p. 295, Jul. 2024. [Online]. Available: https://www.mdpi.com/1999-4893/17/7/295

[25] H. Shirafkan and A. Shameli-Sendi, "Adaptive virtual machine placement: A dynamic approach for energy-efficiency, QoS enhancement, and security optimization," *Cluster Comput.*, vol. 28, no. 1, p. 36, Feb. 2025, doi: 10.1007/s10586-024-04763-2.

[26] E. I. Elsedimy, M. Herajy, and S. M. M. Abohashish, "Energy and QoS-aware virtual machine placement approach for IaaS cloud datacenter," *Neural Comput. Appl.*, vol. 37, no. 4, pp. 2211–2237, Feb. 2025, doi: 10.1007/s00521-024-10872-1.

[27] K. Abbas, J. Hong, N. V. Tu, J.-H. Yoo, and J. W.-K. Hong, "Autonomous DRL-based energy efficient VM consolidation for cloud data centers," *Phys. Commun.*, vol. 55, Dec. 2022, Art. no. 101925, doi: 10.1016/j.phycom.2022.101925.

[28] Y. Wang, Y. Li, T. Wang, and G. Liu, "Towards an energy-efficient data center network based on deep reinforcement learning," *Comput. Netw.*, vol. 210, Jun. 2022, Art. no. 108939, doi: 10.1016/j.comnet.2022.108939.

[29] A. Jayanetti, S. Halgamuge, and R. Buyya, "Multi-agent deep reinforcement learning framework for renewable energy-aware workflow scheduling on distributed cloud data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 35, no. 4, pp. 604–615, Apr. 2024, doi: 10.1109/TPDS.2024.3360448.

[30] G. Cao, "Topology-aware multi-objective virtual machine dynamic consolidation for cloud datacenter," *Sustain. Comput., Informat. Syst.*, vol. 21, pp. 179–188, Mar. 2019, doi: 10.1016/j.suscom.2019.01.004.

[31] M. Biemann, P. A. Gunkel, F. Scheller, L. Huang, and X. Liu, "Data center HVAC control harnessing flexibility potential via real-time pricing cost optimization using reinforcement learning," *IEEE Internet Things J.*, vol. 10, no. 15, pp. 13876–13894, Aug. 2023, doi: 10.1109/JIOT.2023.3263261.

[32] D. Saxena and A. K. Singh, "A proactive autoscaling and energy-efficient VM allocation framework using online multi-resource neural network for cloud data center," *Neurocomputing*, vol. 426, pp. 248–264, Feb. 2021, doi: 10.1016/j.neucom.2020.08.076.

[33] S. Sunil and S. Patel, "Energy-efficient virtual machine placement algorithm based on power usage," *Computing*, vol. 105, no. 7, pp. 1597–1621, Jul. 2023, doi: 10.1007/s00607-023-01152-2.

[34] V. S. Shekhawat, A. Gautam, and A. Thakrar, "Datacenter workload classification and characterization: An empirical approach," in *Proc. IEEE 13th Int. Conf. Ind. Inf. Syst. (ICIIS)*, Dec. 2018, pp. 1–7, doi: 10.1109/ICIINFS.2018.8721402.

[35] B. Predić, L. Jovanovic, V. Simic, N. Bacanin, M. Zivkovic, P. Spalevic, N. Budimirovic, and M. Dobrojevic, "Cloud-load forecasting via decomposition-aided attention recurrent neural network tuned by modified particle swarm optimization," *Complex Intell. Syst.*, vol. 10, no. 2, pp. 2249–2269, Apr. 2024, doi: 10.1007/s40747-023-01265-3.

[36] M. Hosseini Shirvani, "An energy-efficient topology-aware virtual machine placement in cloud datacenters: A multi-objective discrete JAYA optimization," *Sustain. Computing: Informat. Syst.*, vol. 38, Apr. 2023, Art. no. 100856, doi: 10.1016/j.suscom.2023.100856.
[37] H. Feng, Y. Deng, and J. Li, "A global-energy-aware virtual machine placement strategy for cloud data centers," *J. Syst. Archit.*, vol. 116, Jun. 2021, Art. no. 102048, doi: 10.1016/j.sysarc.2021.102048.
[38] A. Tandon and S. Patel, "DBSCAN based approach for energy efficient VM placement using medium level CPU utilization," *Sustain. Comput., Informat. Syst.*, vol. 43, Sep. 2024, Art. no. 101025, doi: 10.1016/j.suscom.2024.101025.
[39] S. Younes, M. Idi, and R. Robbana, "Discrete-time Markov decision process for performance analysis of virtual machine allocation schemes in C-RAN," *J. Netw. Comput. Appl.*, vol. 225, May 2024, Art. no. 103859, doi: 10.1016/j.jnca.2024.103859.
[40] X. Liu, J. Wu, L. Chen, and J. Hu, "A prediction-based multi-objective VM consolidation approach for cloud data centers," *Comput., Mater. Continua*, vol. 80, no. 1, pp. 1601–1631, 2024, doi: 10.32604/cmc.2024.050626.
[41] S. Chen, L. Rui, Z. Gao, Y. Yang, X. Qiu, and S. Guo, "Service migration with edge collaboration: Multi-agent deep reinforcement learning approach combined with user preference adaptation," *Future Gener. Comput. Syst.*, vol. 165, Apr. 2025, Art. no. 107612, doi: 10.1016/j.future.2024.107612.

TAUFIK HIDAYAT (Member, IEEE) received the bachelor's degree from Sekolah Tinggi Ilmu Komputer Poltek, Cirebon, in 2011, and the master's degree from the Department of Electrical Engineering, Universitas Mercu Buana, Indonesia, in 2016. He is currently pursuing the Ph.D. degree in electrical engineering with the Faculty of Engineering, Universitas Indonesia, Depok, Indonesia. Since 2022, he has been a Lecturer with the Department of Computer Engineering, Universitas Wiralodra, and has researched IT value, blockchain, and the Internet of Things. His research interests include machine learning, network security, and mathematical model prediction.

KALAMULLAH RAMLI (Member, IEEE) received the master's degree in telecommunication engineering from the University of Wollongong, Wollongong, NSW, Australia, in 1997, and the Ph.D. degree in computer networks from Universität Duisburg-Essen (UDE), NRW, Germany, in 2003. He has been a Lecturer at Universitas Indonesia (UI) since 1994 and a Professor of computer engineering since 2009. He currently teaches advanced communication networks, embedded systems, object-oriented programming, and engineering and entrepreneurship. His research interests include embedded systems, information and data security, computers and communication, and biomedical engineering.

RUKI HARWAHYU (Member, IEEE) received the B.E. degree in computer engineering from the Universitas Indonesia (UI), Depok, Indonesia, in 2011, the M.E. degree from UI, and the M.Sc. degree in computer and electronic engineering and the Ph.D. degree in electronic and computer engineering from the National Taiwan University of Science and Technology (NTUST), Taipei, Taiwan, in 2013 and 2018 respectively. He is currently an Assistant Professor with the Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia. His current research interests include computer and communication networks and the Internet of Things. He is a member of the editorial boards of two international journals and has organized several international IEEE conferences. He received the Honorary Award from the CTCI Foundation, Taiwan, in 2017.

MUHAMMAD SALMAN (Member, IEEE) received the M.Sc. degree in information technology from Monash University and the Ph.D. degree in information network security from Universitas Indonesia. He was a Vice Chairperson of ID-SIRTII under Indonesia's Ministry of ICT and co-founded Id-CARE.UI, advancing cybersecurity capacity and research. He also led a JICA-supported human resources development project in cybersecurity across Southeast Asia. He is currently a Lecturer and the Head of the Computer Engineering Study Program with the University of Indonesia, specializing in network and information security. He is an Active Member of the IEEE Computer Society, ISSA, ISACA, ISOC, ACM, CSA, and IACSIT.

TEDDY SURYA GUNAWAN (Senior Member, IEEE) received the B.Eng. degree (cum laude) in electrical engineering from the Institut Teknologi Bandung (ITB), Indonesia, in 1998, the M.Eng. degree from Nanyang Technological University, Singapore, in 2001, and the Ph.D. degree from the University of New South Wales (UNSW), Australia, in 2007. He has held esteemed roles, including a Visiting Research Fellow with UNSW, from 2010 to 2021, and an Adjunct Professor with Telkom University from 2022 to 2023. Within International Islamic University Malaysia (IIUM), he was the Head of Department and the Head of Programme Accreditation and Quality Assurance with the Faculty of Engineering, from 2015 to 2016, and from 2017 to 2018, respectively, reinforcing his leadership and expertise in the field. He is currently a Professor with the Department of Electrical and Computer Engineering, IIUM. In addition to his academic and research accomplishments, he holds multiple professional engineering certifications, including the CEng (IET, U.K., 2016), the Insinyur Profesional Utama (PII, Indonesia, 2019), the ASEAN Engineer (2018), the ASEAN Chartered Professional Engineer (2020), the APEC Engineer (2023), the CPEng (Australia, 2024), and the PEng (Malaysia, 2025), reflecting his commitment to professional excellence. Recognized for his contributions to speech and audio processing, biomedical signal processing, image and video processing, and parallel computing, he received IIUM's Best Researcher Award, in 2018 and is listed among the world's top 2% scientists in artificial intelligence and image processing by Elsevier, in 2023 and 2024. He was the Chair of the IEEE Instrumentation and Measurement Society–Malaysia Section.