TELKOMNIKA Telecommunication Computing Electronics and Control
Vol. 24, No. 5, October 2026, pp. 1513~1525
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v24i5.27706

Predictive and fault-tolerant virtual machine migration for energy-efficient cloud data centers

Rakshan G. K., P. Vinothiyalakshmi
Department of Computer Science, Sri Venkateswara College of Engineering, Tamil Nadu, India

Article Info

Article history:

Received Dec 10, 2025
Revised Apr 7, 2026
Accepted May 25, 2026

Keywords:

Failure prediction
Gated recurrent unit-based forecasting
Machine learning
Resource utilization prediction
Virtual machine migration

ABSTRACT

Virtual machine (VM) migration is a key mechanism for improving energy efficiency, service continuity and reliability in cloud data centers. However, conventional migration strategies are largely reactive and fail to account for workload fluctuations and potential failures, often resulting in inefficient resource utilization and increased service-level agreement (SLA) violations. This paper proposes the predictive VM migration manager (PVMM), a unified framework that integrates workload forecasting and failure prediction for proactive migration control. PVMM combines a gated recurrent unit (GRU)-based model for short-term resource prediction with a machine-learning-based failure predictor to identify high-risk VMs. An adaptive dynamic threshold based on energy consumption (ADT-EC) detects host overload conditions, while an improved energy-aware best-fit (IEABF) heuristic selects optimal migration targets. Experimental results demonstrate that PVMM reduces unnecessary migration, improves SLA compliance and enhances overall energy efficiency and system reliability. These findings highlight the effectiveness of predictive, multi-objective migration strategies for next-generation cloud data center management.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Rakshan G. K.
Department of Computer Science, Sri Venkateswara College of Engineering
Pennalur, Sriperumbudur, Tamil Nadu, India.
Email: RakshanGK@outlook.com

1. INTRODUCTION

Virtual machine (VM) migration plays a vital role in addressing these issues by dynamically redistributing workloads to maintain load balance and service continuity [1]–[4]. However, balancing energy efficiency, migration overhead, and reliability remains challenging in large-scale cloud environments [5]–[7]. Conventional migration techniques, including static and dynamic threshold-based policies, respond reactively to overloads without anticipating future workload changes or potential failures [8]–[10]. Such methods can trigger redundant migrations or delay critical actions, increasing service-level agreement (SLA) violations [7].

Advanced approaches leveraging machine learning (ML) have improved decision-making in migration control [6]–[12]. For example, Paulraj *et al.* [11] developed a combined forecast-load-aware strategy, while Mishra *et al.* [8] introduced an interquartile range (IQR)-based model. Yuan *et al.* [12] further improved workload prediction accuracy using long short-term memory (LSTM)-based models. Similarly, Bommala *et al.* [13] applied supervised learning to predict VM failures. However, these methods often optimize a single objective either energy or reliability without jointly addressing both under volatile cloud conditions.

This study presents the predictive VM migration manager (PVMM), an intelligent, fault-tolerant framework that integrates forecasting and failure prediction to enable proactive migration decisions. PVMM

employs a gated recurrent unit (GRU) model for short-term resource utilization forecasting and a ML-based failure predictor to identify high-risk VMs. An adaptive dynamic threshold based on energy consumption (ADT-EC) continuously classifies hosts as underloaded or overloaded, while an improved energy-aware best-fit (IEABF) algorithm selects optimal migration targets, ensuring efficient resource use and SLA compliance.

The proposed PVMM framework introduces a multi-objective, predictive migration strategy that unifies reliability, energy efficiency and operational stability. Unlike conventional reactive models, PVMM anticipates overload and failure conditions before they occur, thereby reducing unnecessary migrations, lowering energy consumption (EC) and improving overall data center resilience. These results contribute to advancing intelligent and sustainable cloud resource management for next-generation computing environments.

2. LITERATURE SURVEY

VM migration is a fundamental mechanism in cloud data centers for improving resource utilization, ensuring service continuity, and reducing EC. Early studies primarily focused on performance modeling and energy-efficient consolidation. Khazaei *et al.* [1] introduced queuing-based analytical models to evaluate performance in cloud environments, while Beloglazov and Buyya [2] proposed adaptive heuristics for dynamic VM consolidation, forming the basis for many energy-aware migration strategies. These foundational works established the importance of balancing resource utilization and SLA compliance.

Subsequent research shifted toward energy-aware and optimization-driven migration techniques. Ma *et al.* [3] provided a comprehensive review of VM migration strategies for energy minimization, while Wang *et al.* [4], Mustafa *et al.* [7] proposed energy- and SLA-aware consolidation approaches to improve operational efficiency. Reinforcement learning and optimization-based methods further advanced this area. For instance, Zhang *et al.* [5] introduced a deep Q-learning model for energy-efficient scheduling, and Gomathi *et al.* [6] explored multi-objective VM placement strategies considering energy and resource utilization.

Adaptive and threshold-based migration mechanisms have also been widely studied. Mishra *et al.* [8] proposed a dynamic threshold-based load balancing technique, while Jiang and Chen [9] developed a self-adaptive VM placement strategy to improve energy efficiency under dynamic workloads. Similarly, Wang *et al.* [10] demonstrated the effectiveness of deep reinforcement learning for energy-efficient VM scheduling. Although these approaches improve adaptability, they largely rely on reactive decision-making based on current system states.

To overcome reactive limitations, predictive and machine-learning-based approaches have been introduced. Paulraj *et al.* [11] proposed a forecast-based migration strategy that integrates workload prediction with migration control. Yuan *et al.* [12] further improved workload prediction accuracy using LSTM-based models, capturing temporal dependencies in large-scale data centers. In addition, Bommala *et al.* [13] developed a machine-learning-based failure prediction model to identify potential system anomalies. These studies highlight the growing role of predictive analytics in enabling proactive migration decisions.

Recent works have also explored intelligent and heuristic-based scheduling frameworks. Abdullah *et al.* [14] proposed a heuristic-based VM consolidation approach, while Buyya *et al.* [15] outlined the vision of autonomic cloud systems capable of self-managing resources. Survey studies [16], [17] emphasize the increasing integration of artificial intelligence and fault-tolerance mechanisms in cloud resource management, while also noting the limitations of existing reactive approaches.

Multi-objective and sustainability-aware migration strategies have gained attention in recent years. Singh *et al.* [18] proposed a QoS-aware task consolidation method to reduce SLA violations, while Khosravi *et al.* [19] introduced a carbon- and energy-aware VM placement strategy for geographically distributed data centers. Similarly, Farahnakian *et al.* [20] applied ant colony optimization for energy-efficient VM consolidation. Comparative analyses, such as Nagma *et al*. [21], further highlight the trade-offs between different migration and consolidation strategies under varying workloads.

More recent studies have emphasized power-aware and communication-aware optimization in VM placement. Arshad *et al.* [22] and Sunil and Patel [23] demonstrated improvements in energy efficiency through dynamic power utilization modeling. In addition, Farzai *et al.* [24] proposed a communication-aware multi-objective optimization model, highlighting the importance of network considerations in VM placement decisions. Accurate power modeling has also been recognized as a key factor in evaluating energy-efficient cloud systems. Lin *et al.* [25] presented a quantitative server power model widely used for estimating EC in cloud data centers, enabling more realistic simulation and evaluation of migration strategies.

Despite these advancements, most existing approaches address either energy efficiency or reliability independently and often rely on reactive or single-objective optimization strategies. However, limited work has focused on integrating workload forecasting, failure prediction, and energy-aware decision-making into a

unified migration framework. To address this gap, this study proposes the PVMM, which integrates GRU-based workload forecasting, machine-learning-based failure prediction, adaptive energy-aware thresholding, and optimized host selection. This unified approach enables proactive, multi-objective VM migration, improving energy efficiency, SLA compliance, and system reliability in dynamic cloud environments.

3. METHODOLOGY

The proposed PVMM introduces a proactive and fault-tolerant VM migration framework that addresses the limitations of reactive approaches. Unlike conventional methods, PVMM integrates workload forecasting, failure prediction and energy-aware decision-making to enable intelligent migration under dynamic cloud conditions. The PVMM architecture consists of six modules (Figure 1), each responsible for a specific stage of predictive migration: data preprocessing, adaptive thresholding, forecasting, failure prediction, risk-based VM selection, host selection and migration execution. These modules operate periodically to ensure continuous monitoring and decision-making.

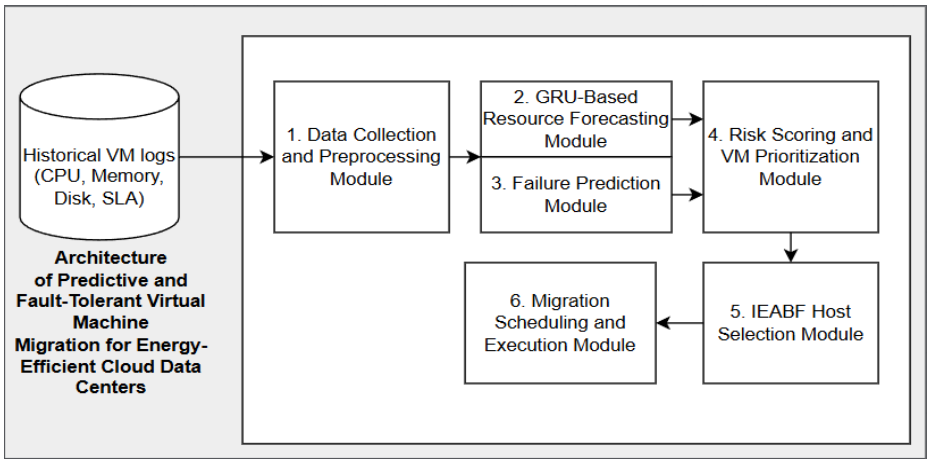

Figure 1. Architecture of the proposed work

3.1. Data collection and preprocessing module

This module collects time-series data for each VM, including central processing unit (CPU), memory, disk utilization, SLA violations and host-level EC. Data is normalized and segmented into sliding windows for model input. Additional features such as task priority, retry count and scheduling delay are extracted for failure prediction.

3.1.1. ADT-EC

The ADT-EC mechanism dynamically classifies host energy states to detect overload conditions. Unlike static thresholds, it adapts based on real-time EC. A host is classified as:

$$E(h) < a \times E_{\max} \Rightarrow \text{BLUE}$$
$$a \times E_{\max} \leq E(h) \leq b \times E_{\max} \Rightarrow \text{GREEN}$$
$$E(h) > b \times E_{\max} \Rightarrow \text{RED}$$

where $E(h)$: current host energy, $E_{\max}$: maximum allowable energy, a, b: adaptive threshold factors ($0.6 \leq a < b \leq 0.9$). These values were selected based on prior threshold-based studies and empirical tuning to balance migration sensitivity and stability. The complete ADT-EC algorithm is presented in Algorithm 1, when a host enters the RED state, migration is triggered. BLUE hosts are preserved to avoid unnecessary migrations.

Algorithm 1. ADT-EC

Input: H: Set of all hosts in the data center
$E(h)$: Current energy consumption of host h
E_{max}: Maximum allowable energy for host h
L_h: Load history of host h
Threshold factors a, b ($0.6 \leq a < b \leq 0.9$)

Output: Host energy state classification Level(h) ∈ {BLUE, GREEN, RED}
Updated migration threshold values for each host
Steps:
1. **Initialize:** Energy Status List (ESL) = Null and Migration Threshold List (MTL) = Null
2. **For each host** $h \in H$:
a. Retrieve $E(h)$and L_h.
b. Compute the average energy utilization $\bar{E}(h)$over the monitoring window.
3. **Classify energy consumption level:**
i. **If** $E(h) < a \times E_{max}$:
-Set Level(h) = BLUE // Host is idle or underutilized.
ii. **Else if** $a \times E_{max} \leq E(h) \leq b \times E_{max}$:
-Set Level(h) = GREEN // Host is in normal range.
iii. **Else if** $E(h) > b \times E_{max}$:
-Set Level(h) = RED // Host is overloaded.
4. **Adjust migration threshold (MT)** based on the classified level:
-For **BLUE,** increase MT to avoid unnecessary migrations.
-For **GREEN,** maintain MT at nominal value.
-For **RED,** lower MT to trigger immediate VM migration.
5. Update **ESL** and **MTL** with Level(h)and corresponding threshold.
6. Identify hosts as:
Overloaded if Level(h) = RED
Underutilized if Level(h) = BLUE
7. **End For**
8. **Return** [ESL,MTL]to the VM Selection Module.
9. **End**

3.2. GRU-based resource forecasting module

A GRU model is used to predict short-term VM resource utilization (CPU, memory, disk). GRU is selected due to its efficiency in capturing temporal dependencies with lower computational overhead than LSTM. The model is trained on historical workload traces and predicts future utilization to enable proactive overload detection.

3.3. Failure prediction module

A supervised learning model (random forest (RF) / extreme gradient boosting (XGBoost)) predicts VM failure probability:

$$\mathrm{FR}(v) \in [0,1]$$

Feature vectors include CPU pressure levels, task retry frequency, resource queue length, recent hardware error logs, and scheduling anomaly indicators. It is trained on labelled cloud trace datasets to classify VMs into low-risk and high-risk categories based on likelihood of failure.

3.4. Overload risk scoring and VM prioritization module

For every VM v , a composite risk score is computed:

$$S_v = x * FR(v) + y * \hat{\mathrm{U}}_v + \mathrm{z} * V_{SLA}(v) \quad (1)$$

where, $FR(v)$ is the predicted failure risk, $\hat{\mathrm{U}}_v$ is the forecasted resource utilization (average of CPU, memory (MEM), and disk (DISK)), $V_{SLA}(v)$ is the recent SLA violation frequency, $[x, y, z]$ are the empirically tuned weighting parameters. The weights are determined through empirical tuning and sensitivity analysis to balance reliability and energy efficiency. The VMs are sorted by descending score S_v and the candidates are selected for migration.

3.4.1. Correlation-based VM candidate selection (SS-CAU)

The SS-CAU strategy selects VMs contributing most to overload using correlation:

$$R_v = \mathrm{corr}(U_v^{\mathrm{CPU}}, U_h^{\mathrm{CPU}})$$

VMs with $R_v > 0.6$ are selected and sorted by ascending utilization to minimize migration overhead. The complete SS-CAU procedure is presented in Algorithm 2. This ensures targeted load reduction with minimal data transfer cost.

Algorithm 2. SS-CAU

Input: H_{over}: Set of overloaded hosts identified by ADT-EC
$V(h)$: Set of VMs residing on host h
U_v^{CPU}: CPU utilization history of VM v
U_h^{CPU}: CPU utilization history of host h
Correlation threshold $r_{min} = 0.6$
Output: Candidate VM list for migration C_{VM}
Steps:
1. **Initialize:** $C_{VM} = \emptyset$ and R_v is the Correlation coefficient for VM v
2. **For each overloaded host** $h \in H_{over}$:
a. Retrieve all VMs $V(h)$associated with host h.
b. For each VM $v \in V(h)$:
i. Compute the correlation coefficient R_v between VM and host CPU utilization using regression:

$$R_v = \text{corr}(U_v^{CPU}, U_h^{CPU})$$

ii. **If** $R_v > r_{min}$:
- Add v to candidate list C_{VM}.
c. **End For**
3. **Sort** all candidate VMs $v \in C_{VM}$in **ascending order of current CPU utilization** $U_v^{CPU}(t)$.
4. **Select** VMs sequentially for migration starting from lowest CPU utilization, as smaller workloads migrate faster and with less data transfer overhead.
5. Continue migration **until the host's utilization** U_h^{CPU} **drops below the adaptive threshold** determined by the ADT-EC algorithm.
6. **End For**
7. **Return** the final candidate list C_{VM} to the **Host Selection (IEABF)** module.
8. **End**

3.5. IEABF host selection module

The IEABF algorithm selects optimal destination hosts. For each VM:

$$\Delta E(h) = E_{\text{post}}(h) - E_{\text{current}}(h) \quad (2)$$

The host with the minimum $\Delta E(h)$value and sufficient available capacity is selected as the destination. Algorithm 3 summarizes the IEABF host selection procedure. This best-fit heuristic ensures optimal load distribution and reduced power overhead while maintaining SLA compliance. By prioritizing energy efficiency and operational stability, IEABF enhances the overall effectiveness of the PVMM framework, leading to improved resource utilization, lower migration frequency and extended data center lifespan. The algorithm operates with polynomial time complexity relative to the number of hosts and VMs, making it suitable for scalable cloud environments.

Algorithm 3. IEABF host selection

Input: C_{VM}: List of VMs selected for migration (from SS-CAU)
H: Set of all available hosts in the data center
$E(h)$: Current energy consumption of host h
E_{max}: Maximum allowable energy for host h
C_h: Available computing capacity (CPU, memory, disk) of host h
$V_{SLA}(h)$: Current SLA violation rate of host h
Thresholds for utilization and SLA compliance
Output: Optimized VM-to-host mapping list $M = \{(v, h_{target})\}$
Steps:
1. **Initialize:** $M = \emptyset$ and $H_{valid} = \emptyset$
2. **Preprocessing:**
a. Exclude hosts that are **overloaded or underloaded** according to the ADT-EC classification.
b. Retain only hosts satisfying $E(h) < E_{max}$and $V_{SLA}(h) <$ threshold.
c. Add all such hosts to H_{valid}.
3. **Sort** VMs in $C_{VM}$by **descending CPU utilization,** prioritizing high-demand VMs for placement first.
4. **For each VM** $v \in C_{VM}$:
a. Initialize $\Delta E_{min} = \infty$; $h_{best} = \text{null}$.
b. **For each candidate host** $h \in H_{valid}$:
i. Check resource capacity constraint: $C_h \geq \text{ResourceDemand}(v)$.
ii. Estimate the expected post-placement energy:

$$E_{post}(h) = E_{current}(h) + \text{EnergyIncrement}(v)$$

iii. Compute energy difference:

$$\Delta E(h) = E_{post}(h) - E_{current}(h)$$

iv. **If** $\Delta E(h) < \Delta E_{min}$and capacity constraints are satisfied:
- Update $\Delta E_{min} = \Delta E(h)$

```
        - Set h_best = h
c. Assign VM v to h_best and update host energy state.
d. Add mapping (v, h_best) to M.
e. Remove h_best from H_valid if capacity becomes saturated.
5.      End For
6.      Return final mapping list M for migration execution.
7.      End
```

3.6. Migration scheduling and execution module

This module coordinates migration decisions and execution.

3.6.1. Migration triggering

A migration readiness index (MRI) is defined:

$$\mathrm{MRI}(v) = \alpha \cdot R_{\mathrm{forecast}}(v) + \beta \cdot F_{\mathrm{risk}}(v) + \gamma \cdot E_{\mathrm{state}}(h)$$

where $R_{\mathrm{forecast}}(v)$ represents the forecasted resource demand growth rate, $F_{\mathrm{risk}}(v)$ the predicted failure probability and $E_{\mathrm{state}}(h)$ the current energy level of the host h. The weighting parameters $(\alpha, \beta, \gamma)$are empirically tuned to prioritize either failure avoidance or energy efficiency. Migration is triggered only when the computed MRI exceeds a defined activation threshold, preventing unnecessary VM relocations.

3.6.2. Migration execution

Once a migration decision is approved, live VM migration is performed using a pre-copy mechanism, where memory pages are iteratively transferred to the destination host while the VM remains active, minimizing service downtime. Prior to migration, the scheduler verifies the suitability of the destination host based on capacity, energy constraints and SLA requirements defined by the IEABF module. The migration process consists of three phases: (i) pre-copy, where memory pages are progressively transferred; (ii) stop-and-copy, where the VM is briefly paused to transfer remaining state; and (iii) state update, where VM mappings are updated and post-migration energy $E_{post}$is recalculated.

To ensure network stability during concurrent migrations, the scheduler applies adaptive bandwidth control, throttling migration traffic when network utilization exceeds predefined thresholds. After migration, validation checks are performed to ensure system consistency and SLA compliance. These include verification of VM state integrity and monitoring of response time and throughput. If performance degradation exceeds acceptable limits, a rollback mechanism is triggered.

3.6.3. Migration cost and overhead minimization

Migration overhead is evaluated using the cost function:

$$C_{\mathrm{mig}}(v, h_{\mathrm{src}}, h_{\mathrm{dst}}) = \omega_1 T_{\mathrm{mig}} + \omega_2 B_{\mathrm{mig}} + \omega_3 \Delta E(h_{\mathrm{dst}})$$

where T_{mig} is migration time, B_{mig} is bandwidth usage and $\Delta E(h_{\mathrm{dst}})$ is the incremental energy cost of the destination host. The weighting factors $(\omega_1, \omega_2, \omega_3)$ tuned to balance latency and energy efficiency. Migration is executed only when the benefit-to-cost ratio exceeds a defined threshold, preventing unnecessary migrations.

3.6.4. Summary of operation

The scheduler integrates decisions from ADT-EC, SS-CAU and IEABF to perform coordinated VM migration. It handles migration triggering, execution and validation while minimizing overhead through predictive scheduling and network-aware control. This integrated approach ensures reduced downtime, improved energy efficiency and sustained SLA compliance.

4. EXPERIMENTAL SETUP

The proposed PVMM framework was evaluated using both real-world cloud workload traces and simulation-based environments to assess its performance in energy efficiency, migration frequency and fault-tolerant prediction accuracy. The framework integrates energy-aware migration control, GRU-based resource forecasting and machine-learning-driven failure prediction to enable proactive and intelligent migration decisions within dynamic cloud infrastructures.

4.1. Simulation environment

Experiments were conducted using the CloudSim 3.0.3 toolkit, a well-established simulation framework for evaluating energy-aware resource management and VM migration strategies in cloud data centers. The simulated environment comprised 100 heterogeneous physical hosts modeled after SPECpower benchmark servers with HP ProLiant ML110 G4 (Intel Xeon 3040, 2 cores, 1.86 GHz, 4 GB RAM) and HP ProLiant ML110 G5 (Intel Xeon 3075, 2 cores, 2.66 GHz, 4 GB RAM) specifications. The power consumption profiles of these hosts were derived from established server power modeling approaches in cloud data centers [25], which are widely used to approximate energy utilization based on workload intensity. Each host supported 10 to 15 VMs with variable resource demands and load patterns derived from PlanetLab workload traces, ensuring realistic temporal utilization fluctuations that emulate real-world operational conditions.

Each simulation represented 24 hours of continuous operation, during which the CloudSim energy model was extended to compute dynamic EC states for each host, represented as $E_{current}$ and E_{post}, to capture pre- and post-migration energy variations. The ADT-EC mechanism was integrated to classify host energy states and dynamically adjust migration thresholds based on load history and power utilization trends.

The GRU-based resource forecasting and RF-based failure prediction models were pre-trained externally in Python using the Google Cluster, LANL Mustang and Trinity traces and then embedded within CloudSim for inference during runtime. Predictions were exchanged between the ML models and the simulator through a lightweight JavaScript Object Notation (JSON)-based interface, enabling CloudSim to request forecast and failure probability values at each scheduling interval. The inference latency of the ML models is assumed to be negligible in the simulation environment, as predictions are pre-computed and invoked at discrete scheduling intervals rather than executed as real-time streaming inference. This abstraction allows the study to focus on migration decision effectiveness rather than runtime ML overhead.

To ensure reproducibility and reduce stochastic bias, each experiment was repeated five times with different random seeds and average results were reported for all evaluation metrics. The framework is designed to scale with increasing numbers of hosts and VMs, as the modular architecture distributes forecasting, prediction and scheduling tasks independently. However, computational overhead associated with large-scale inference and migration coordination is not explicitly modeled in CloudSim and thus scalability is evaluated primarily in terms of system behavior rather than execution time.

4.2. Dataset and workload traces

Three publicly available cloud workload datasets were utilized for training and validating the predictive models integrated into the PVMM framework. The Google Cluster Trace was employed to represent production-scale, heterogeneous workloads typical of large cloud infrastructures. The LANL Mustang and Trinity traces were used to capture high-performance computing workload characteristics, including varying task complexities, execution durations and resource contention behaviors. Each dataset contained detailed information on CPU utilization, memory consumption, disk input/output (I/O), task priority, retry count and exit codes.

All datasets were cleaned and preprocessed to remove duplicates, incomplete records and anomalies, following standardized procedures described in recent machine-learning-based failure prediction studies [13]. The preprocessed data were partitioned into training and testing sets in the ratio of 70:30 for forecasting and 80:20 for failure prediction. They were then used to develop the GRU-based resource forecasting and RF-based failure prediction models in TensorFlow v2.12 and Scikit-learn v1.4, respectively. The trained models were then exported and interfaced with CloudSim 3.0.3 to provide real-time predictions during simulation.

For simulation workloads, PlanetLab traces were employed to generate time-varying utilization inputs for the VMs hosted on each simulated server. These traces effectively reproduce dynamic cloud conditions by introducing realistic fluctuations in CPU and memory demand. This combined use of real-world trace data for model training and PlanetLab-based workload emulation ensured that the simulation environment accurately reflected both operational unpredictability and predictive decision-making behavior within large-scale cloud data centers.

4.3. Model training and parameter configuration

The resource forecasting model was implemented using a GRU neural network developed in TensorFlow v2.12. The GRU was trained on 70% of the historical time-series data, which included CPU, memory and disk utilization metrics, while the remaining 30% was reserved for testing and validation. Each input sequence was constructed using a 10-step sliding window, corresponding to 5-minute sampling intervals, enabling the model to capture temporal dependencies across short-term workload fluctuations. The network architecture consisted of two hidden layers with 64 neurons each and was optimized using the Adam optimizer with a learning rate of 0.001. The mean squared error (MSE) loss function was employed to minimize prediction error during training. This configuration allowed the GRU model to effectively learn

dynamic utilization patterns and forecast near-future workload trends, facilitating proactive migration decisions before overload conditions occurred.

The failure prediction model was implemented using supervised ML techniques in Scikit-learn v1.4, employing an 80:20 split for training and testing. The feature set included key behavioral and system-level indicators such as task retry count, resource contention index, SLA violation frequency and CPU queue length. Multiple classifiers, including RF and XGBoost, were evaluated to determine optimal performance. Among these, the RF model, configured with 200 estimators, demonstrated superior predictive capability, achieving an F1-score of 0.94, consistent with prior findings in high-confidence computing Bommala *et al.* [13]. The failure probability scores generated by this model were integrated into the PVMM framework to prioritize migration of high-risk VMs, thereby enhancing system fault tolerance and service continuity.

4.4. Performance metrics

The performance of the proposed framework was assessed using a set of quantitative metrics that collectively measure energy efficiency, reliability and operational stability. EC represents the total energy utilized by all active hosts during simulation. SLA violation denotes the ratio of SLA breaches such as missed response deadlines or latency violations to the total number of executed tasks. The Number of VM migrations reflects the frequency of live migration events per simulation cycle and serves as an indicator of system stability. Failure prediction accuracy was quantified through precision, recall and F1-score metrics obtained from the ML-based failure predictor. Finally, average CPU utilization represents the mean host utilization level after migration, providing insight into overall load balancing efficiency.

4.5. Baseline comparison methods

To validate the effectiveness of the proposed framework, comparative experiments were conducted against four established VM migration approaches: static threshold-based migration [2], dynamic IQR-based migration [8], combined forecast load-aware (CF-LA) migration [11] and the EABF algorithm [3]. These baseline methods encompass a spectrum of strategies ranging from static and dynamically adjusted thresholds to predictive, load-aware and energy-optimized migration policies, enabling a comprehensive evaluation of the improvements achieved by the proposed predictive and fault-tolerant migration framework.

4.6. Threats to validity

The experimental setup has certain limitations. CloudSim abstracts low-level system factors such as network latency, hardware heterogeneity and virtualization overhead, which may affect real-world performance. ML inference latency is not explicitly modeled, as predictions are pre-trained and invoked during simulation. Additionally, the datasets used (Google Cluster and LANL traces) may not capture all workload variations, potentially affecting generalization. Scalability is evaluated at the system level, as computational overhead of large-scale inference and migration coordination is not fully represented in the simulation. Despite these limitations, the setup provides a consistent and reproducible basis for evaluating predictive and energy-aware VM migration.

5. PERFORMANCE EVALUATION

The performance of the PVMM was analyzed using the setup described above, with emphasis on energy efficiency, migration frequency, SLA compliance and failure prediction accuracy. The data shown is from 5 runs each with the mean values shown.

5.1. Evaluation objectives

The experimental evaluation was designed to comprehensively assess the effectiveness of the proposed framework across its key functional components. Specifically, the objectives were to validate the ADT-EC mechanism in minimizing unnecessary migrations and stabilizing host workloads. To evaluate the accuracy of the GRU-based forecasting model under dynamically varying cloud workloads. To measure the capability of the failure prediction model in identifying high-risk VMs before potential failure events. And to quantify the improvements in energy efficiency and SLA compliance achieved through the IEABF host selection algorithm. All reported results represent mean values over five independent simulation runs. The observed performance trends remained consistent across runs with low variance, indicating stability of the proposed framework.

5.2. EC analysis

Figure 2 illustrates average EC across migration strategies. The ADT-EC+IEABF combination reduced total energy usage by approximately 15-18% relative to dynamic IQR-based methods and up to 22%

compared with static threshold migration. These findings align with Ma *et al.* [3], where adaptive energy thresholds achieved an average 15.49% reduction. The improvement stems from dynamic host reclassification to BLUE, GREEN, and RED states, which minimizes redundant migrations and consolidates workloads only when EC exceeds the adaptive limit of $b \cdot E_{\max}$. This improvement is primarily due to proactive consolidation enabled by workload forecasting, which prevents unnecessary host activation and reduces energy spikes.

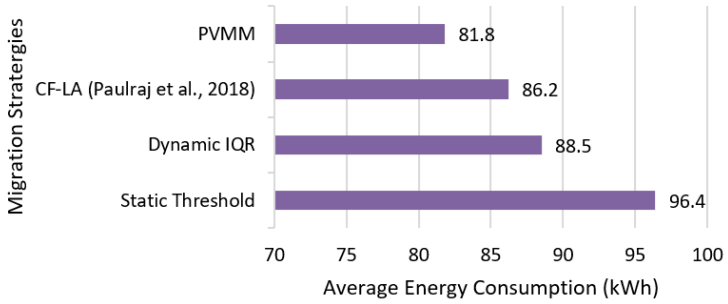

Figure 2. Comparison of average EC (in kWh) among different VM migration strategies

5.3. Migration frequency and stability

As shown in Figure 3, the proposed method reduced the number of VM migrations by approximately 80% compared with dynamic threshold techniques and by over 83% compared with static policies, closely matching the 83.32% reduction observed by Ma *et al.* [3]. The SS-CAU correlation-based selection algorithm contributed significantly by prioritizing VMs most correlated with host overload while minimizing data transfer overhead. This reduced oscillatory migrations and improved host utilization stability. The reduction is further attributed to predictive migration timing, which avoids reactive oscillations commonly observed in threshold-based approaches.

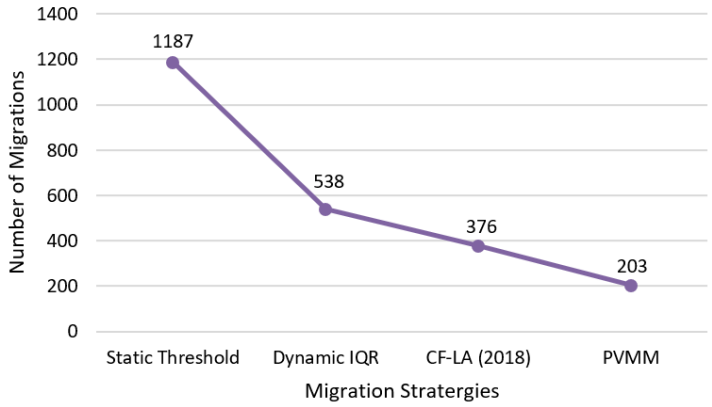

Figure 3. Comparison of the number of VM migrations per 24-hour simulation period across different migration stratergies

5.4. SLA violation rate (SLAV)

The SLAV, defined as the percentage of tasks exceeding performance constraints, was reduced by 7% to 9% compared with existing approaches as shown in Figure 4. The improvement results from proactive migration triggers enabled by forecasted utilization and failure risk scores, ensuring that hosts remain below the safe energy threshold $E_{\max}$ after reallocation. The IEABF host selection process further maintained SLA adherence by preventing overload propagation during consolidation. By anticipating overload conditions, PVMM reduces delayed migrations, which are a primary cause of SLA violations in reactive systems.

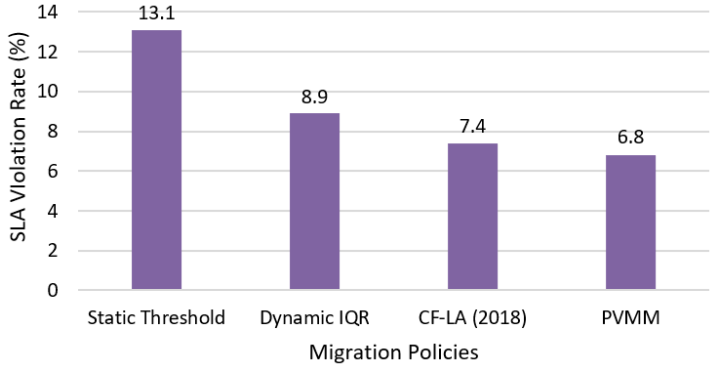

Figure 4. Shows the SLA violation rates for various migration policies

5.5. Failure prediction performance

The RF and XGBoost models, trained on the Google, Mustang and Trinity traces, demonstrated superior predictive accuracy compared to baseline classifiers, as shown in Figure 5. Among them, the RF model achieved an accuracy of 94.1%, precision of 92.8%, recall of 94.8% and an F1-score of 93.6%. These results validate the effectiveness of multi-trace training in capturing diverse workload behaviors, enabling the model to accurately anticipate VM-level failures. The resulting failure risk predictions facilitated pre-emptive migrations, thereby reducing potential service interruptions and improving overall system resilience.

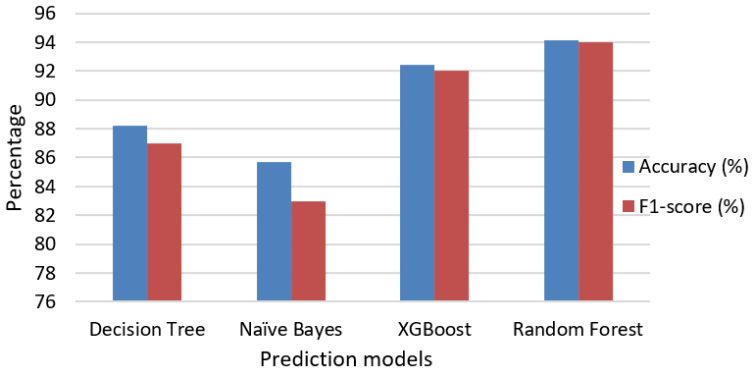

Figure 5. Shows the comparison of failure prediction model performance across classifiers

5.6. Resource forecasting accuracy

The GRU-based forecasting module outperformed traditional time-series models such as autoregressive integrated moving average (ARIMA) and Holt-Winter's exponential smoothing. The mean absolute percentage error (MAPE) was 3.8% for CPU and 4.6% for memory utilization predictions, compared with 7% to 9% for ARIMA. Accurate short-term forecasting facilitated timely activation of the migration scheduler, improving load balance and preventing overload propagation. This improvement is due to the GRU model's ability to capture temporal dependencies in workload patterns, enabling more accurate short-term predictions.

5.7. Comparative performance summary

Table 1 presents a comparative summary of the proposed PVMM framework against three baseline strategies-Static Threshold [2], Dynamic IQR-based migration [8] and CF-LA migration [11]. The results clearly demonstrate the superior performance of PVMM across all evaluated metrics. Specifically, PVMM achieves the highest energy reduction of 18.2%, outperforming CF-LA and IQR-based approaches by 4.4% and 6.7%, respectively. The framework also reduces the frequency of VM migrations by 83.3% compared to static threshold methods, indicating improved system stability and fewer redundant migrations.

In terms of reliability, PVMM attains a 6.8% SLA violation rate, the lowest among all compared methods, while maintaining a failure prediction accuracy of 94.1% and a MAPE of only 3.8%. These results confirm that the integration of predictive analytics, adaptive thresholding and energy-aware placement in PVMM leads to significant gains in energy efficiency, fault tolerance and SLA compliance compared with existing migration schemes.

Table 1. Comparative performance summary of VM migration and prediction strategies

Metric	Static threshold [2]	Dynamic IQR [8]	CF-LA [11]	Proposed (PVMM)
0	0	11.5	13.8	18.2
Migration reduction (%)	0	55.2	68.7	83.3
SLAV (%)	13.1	8.9	7.4	6.8
Failure prediction accuracy (%)	-	-	89.5	94.1
Forecast error (MAPE, %)	-	9.2	7.1	3.8

5.8. Discussion

The results demonstrate that integrating predictive analytics with energy-aware migration significantly improves system efficiency and reliability. The GRU-based forecasting model enhances migration timing by capturing temporal workload patterns, while failure prediction enables proactive handling of high-risk VMs, reducing SLA violations. The reduction in migration frequency and EC indicates a favorable cost-benefit trade-off, where the gains from reduced energy usage outweigh the computational overhead of prediction models. However, in some cases, prediction inaccuracies may result in unnecessary migrations, though their overall impact remains limited.

Although the weighting parameters used in risk scoring and migration decision-making were empirically tuned, a detailed parameter sensitivity analysis was not exhaustively explored in this study. Preliminary observations indicate that moderate variations in weights do not significantly alter overall performance trends. A comprehensive sensitivity and ablation study will be considered in future work to further quantify the impact of parameter selection.

While the evaluation focuses on threshold-based and predictive baselines, reinforcement learning and metaheuristic approaches have also shown potential in VM migration optimization. Incorporating such methods into comparative analysis is part of future work. It is important to note that the results are based on simulation using CloudSim, which does not fully capture real-world factors such as network latency and hardware variability. Therefore, the findings indicate promising performance, though real-world validation is required. Additionally, model retraining may be necessary under dynamic workload changes to maintain prediction accuracy.

6. CONCLUSION

This study presented the PVMM, a unified framework for improving energy efficiency and reliability in cloud data centers through predictive and energy-aware migration. By integrating GRU-based workload forecasting, machine-learning-based failure prediction and adaptive threshold-driven migration control, PVMM enables proactive decision-making beyond conventional reactive strategies. Experimental results demonstrate that PVMM achieves up to 18% reduction in EC and over 80% reduction in migration frequency, while also improving SLA compliance. These improvements are primarily driven by accurate workload forecasting, early failure detection and energy-aware host selection, which collectively reduce unnecessary migrations and prevent overload propagation. PVMM demonstrates that predictive, risk-aware migration can significantly advance intelligent cloud resource management, offering a practical approach to balancing energy efficiency, reliability and system stability. However, the evaluation is limited to simulation-based environments and does not explicitly account for real-world factors such as network latency, hardware heterogeneity and large-scale inference overhead. Additionally, model performance may vary under unseen workload distributions. Future work will explore real-world deployment, scalability optimization and integration with reinforcement-learning-based scheduling and containerized cloud environments to further enhance adaptability and robustness.

FUNDING INFORMATION

Authors state there is no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Rakshan G. K.	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓	
P. Vinothiyalakshmi	✓	✓		✓			✓			✓		✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Author states no conflict of interest.

DATA AVAILABILITY

Not available.

REFERENCES

[1] H. Khazaei, J. Misic, and V. B. Misic, "Performance analysis of cloud computing centers using M/G/m/m+r queuing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 5, pp. 936–943, May 2012, doi: 10.1109/TPDS.2011.199.

[2] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, Sep. 2012, doi: 10.1002/cpe.1867.

[3] Z. Ma, D. Ma, M. Lv, and Y. Liu, "Virtual machine migration techniques for optimizing energy consumption in cloud data centers," *IEEE Access*, vol. 11, pp. 86739–86753, 2023, doi: 10.1109/ACCESS.2023.3305268.

[4] J. Wang, H. Gu, J. Yu, Y. Song, X. He, and Y. Song, "Research on virtual machine consolidation strategy based on combined prediction and energy-aware in cloud computing platform," *Journal of Cloud Computing*, vol. 11, no. 1, p. 50, Sep. 2022, doi: 10.1186/s13677-022-00309-2.

[5] Q. Zhang, M. Lin, L. T. Yang, Z. Chen, S. U. Khan, and P. Li, "A double deep q-learning model for energy-efficient edge scheduling," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 739–749, Sep. 2019, doi: 10.1109/TSC.2018.2867482.

[6] B. Gomathi *et al.*, "Multi-objective optimization of energy aware virtual machine placement in cloud data center," *Intelligent Automation & Soft Computing*, vol. 33, no. 3, pp. 1771–1785, 2022, doi: 10.32604/iasc.2022.024052.

[7] S. Mustafa, K. Bilal, S. U. R. Malik, and S. A. Madani, "SLA-aware energy efficient resource management for cloud environments," *IEEE Access*, vol. 6, pp. 15004–15020, 2018, doi: 10.1109/ACCESS.2018.2808320.

[8] S. K. Mishra, M. A. Khan, B. Sahoo, D. Puthal, M. S. Obaidat, and K. Hsiao, "Time efficient dynamic threshold-based load balancing technique for Cloud Computing," in *2017 International Conference on Computer, Information and Telecommunication Systems (CITS)*, Jul. 2017, pp. 161–165, doi: 10.1109/CITS.2017.8035327.

[9] H.-P. Jiang and W.-M. Chen, "Self-adaptive resource allocation for energy-aware virtual machine placement in dynamic computing cloud," *Journal of Network and Computer Applications*, vol. 120, pp. 119–129, Oct. 2018, doi: 10.1016/j.jnca.2018.07.011.

[10] B. Wang, F. Liu, and W. Lin, "Energy-efficient VM scheduling based on deep reinforcement learning," *Future Generation Computer Systems*, vol. 125, pp. 616–628, Dec. 2021, doi: 10.1016/j.future.2021.07.023.

[11] G. J. L. Paulraj, S. A. J. Francis, J. D. Peter, and I. J. Jebadurai, "A combined forecast-based virtual machine migration in cloud data centers," *Computers & Electrical Engineering*, vol. 69, pp. 287–300, Jul. 2018, doi: 10.1016/j.compeleceng.2018.01.012.

[12] H. Yuan, J. Bi, S. Li, J. Zhang, and M. Zhou, "An improved LSTM-based prediction approach for resources and workload in large-scale data centers," *IEEE Internet of Things Journal*, vol. 11, no. 12, pp. 22816–22829, Jun. 2024, doi: 10.1109/JIOT.2024.3383512.

[13] H. Bommala, U. M. V., R. Aluvalu, and S. Mudrakola, "Machine learning job failure analysis and prediction model for the cloud environment," *High-Confidence Computing*, vol. 3, no. 4, p. 100165, Dec. 2023, doi: 10.1016/j.hcc.2023.100165.

[14] M. Abdullah, K. Lu, P. Wieder, and R. Yahyapour, "A heuristic-based approach for dynamic VMs consolidation in cloud data centers," *Arabian Journal for Science and Engineering*, vol. 42, no. 8, pp. 3535–3549, Aug. 2017, doi: 10.1007/s13369-017-2580-5.

[15] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, Jun. 2009, doi: 10.1016/j.future.2008.12.001.

[16] P. Kumari and P. Kaur, "A survey of fault tolerance in cloud computing," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 10, pp. 1159–1176, Dec. 2021, doi: 10.1016/j.jksuci.2018.09.021.

[17] B. Prabha, K. Ramesh, and P. N. Renjith, "A Review on Dynamic Virtual Machine Consolidation Approaches for Energy-Efficient Cloud Data Centers," *Algorithms for Intelligent Systems*, pp. 761–780, 2021, doi: 10.1007/978-981-15-8530-2_60.

[18] P. Singh, V. Prakash, G. Bathla, and R. K. Singh, "QoS aware task consolidation approach for maintaining SLA violations in

cloud computing," *Computers and Electrical Engineering*, vol. 99, p. 107789, Apr. 2022, doi: 10.1016/j.compeleceng.2022.107789.
[19] A. Khosravi, L. L. H. Andrew, and R. Buyya, "Dynamic VM placement method for minimizing energy and carbon cost in geographically distributed cloud data centers," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 2, pp. 183–196, Apr. 2017, doi: 10.1109/TSUSC.2017.2709980.
[20] F. Farahnakian *et al.*, "Using ant colony system to consolidate VMs for green cloud computing," *IEEE Transactions on Services Computing*, vol. 8, no. 2, pp. 187–198, Mar. 2015, doi: 10.1109/TSC.2014.2382555.
[21] Nagma, J. Singh, and J. Sidhu, "Comparative analysis of VM consolidation algorithms for cloud computing," *Procedia Computer Science*, vol. 167, pp. 1390–1399, 2020, doi: 10.1016/j.procs.2020.03.350.
[22] U. Arshad, M. Aleem, G. Srivastava, and J. C.-W. Lin, "Utilizing power consumption and SLA violations using dynamic VM consolidation in cloud data centers," *Renewable and Sustainable Energy Reviews*, vol. 167, p. 112782, Oct. 2022, doi: 10.1016/j.rser.2022.112782.
[23] 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.
[24] S. Farzai, M. H. Shirvani, and M. Rabbani, "Multi-objective communication-aware optimization for virtual machine placement in cloud datacenters," *Sustainable Computing: Informatics and Systems*, vol. 28, p. 100374, Dec. 2020, doi: 10.1016/j.suscom.2020.100374.
[25] W. Lin, W. Wu, H. Wang, J. Z. Wang, and C.-H. Hsu, "Experimental and quantitative analysis of server power model for cloud data centers," *Future Generation Computer Systems*, vol. 86, pp. 940–950, Sep. 2018, doi: 10.1016/j.future.2016.11.034.

BIOGRAPHIES OF AUTHORS

Rakshan G. K. is presently studying as a student in the Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Tamil Nadu, India. He is currently in fourth year of his Bachelor's degree in Computer Science and Engineering. His research interests include cloud computing, machine learning, deep learning, and distributed systems. He can be contacted at email: RakshanGK@outlook.com.

P. Vinothiyalakshmi is presently working as a Professor in the Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Tamil Nadu, India. She obtained Ph.D. degree from Anna University during the year 2021 for her thesis entitled "Credibility Based Multi-Attribute Combinative Double Auction for Resource Allocation in Cloud Computing". Having 18 years of experience in teaching and her research interests include cloud computing, machine learning, deep learning, computer networks and data structures. She can be contacted at email: vlakshmi@svce.ac.in.