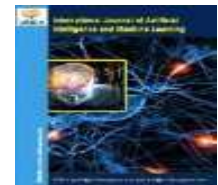




International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Adaptive Risk-Aware Energy and Integrity Optimization for Sustainable Cloud Resource Management

Sachin H. Patel¹, Amit Nayak²

¹Department of Computer Engineering, Chandubhai S. Patel Institute of Technology (CSPIT), Faculty of Technology & Engineering, Charotar University of Science and Technology (CHARUSAT), Gujarat, India.

²Department of Computer Science and Engineering, Devang Patel Institute of Advanced Technology and Research (DEPSTAR), Faculty of Technology & Engineering, Charotar University of Science and Technology (CHARUSAT), Gujarat, India.

*Corresponding Author: Sachinpatel.it@gmail.com, 21drdce002@charusat.edu.in, sachinpatel.dit@charusat.ac.in

Abstract

Cloud data centres must reduce energy use without weakening the controls that protect data integrity. This paper presents an adaptive risk-aware resource-management framework that couples validation intensity, workload prediction uncertainty, VM consolidation, and migration decisions within a five-term objective covering energy, residual integrity exposure, SLA stress, validation overhead, and migration cost. The revised methodology defines system entities, state and decision variables, placement and capacity constraints, risk and uncertainty construction, normalization, candidate evaluation, and stability controls. The controller was evaluated in a reproducible 720-hour controlled simulation using 100 normalized hosts. To improve robustness beyond a single scenario, 30 independent stochastic runs were executed with identical paired inputs for all policies, and 95% confidence intervals, paired statistical tests, ablation experiments, and coefficient-sensitivity analysis were added. Across the 30 runs, the proposed adaptive policy used 101.34% relative energy, achieved an integrity-exposure index of 6.96, an SLA-violation rate of 13.55%, and validation overhead of 0.77%. Compared with the prior joint policy, it reduced integrity exposure by 18.4% and reduced SLA violations by 0.26 percentage points, while increasing relative energy by 0.32 percentage points. Compared with fixed validation, it reduced energy and validation overhead while accepting higher residual exposure. The results demonstrate a controllable and statistically repeatable trade-off rather than a single-objective optimum, and the paper provides an implementation mapping for AWS-style and orchestrated cloud environments.

Keywords

Cloud computing; energy efficiency; data integrity; adaptive validation; virtual machine consolidation; workload prediction; anomaly detection; multi-objective optimization; SLA; sustainable computing

1. Introduction

A cloud data centre has to respond quickly to changing demand, but it must also keep energy use, service quality and security risk under control. Virtualization helps because workloads can be consolidated onto fewer physical hosts. The difficulty is that very aggressive consolidation may increase contention, trigger more migrations and create additional exposure while VM state is being moved. Security mechanisms add a different kind of cost. Integrity verification, log processing, encryption and attestation all consume computing resources, so they also influence the energy profile of the data centre. For this reason, a controller that looks only at host utilization can underestimate the cost of protection. At the other extreme, running every integrity check at a fixed high level can waste resources during periods in which the observed risk is small.

Our previous work developed this problem step by step. The 2023 study reviewed methods for improving cloud energy efficiency and security and pointed to the need to consider the two objectives together [1]. AegisCloud then focused on the security side by using adversarial contrastive Transformer-based analysis of cloud logs [2]. In 2026, we presented an integrated architecture that combined integrity-aware consolidation, machine-learning-based prediction, AWS implementation and multi-objective energy-integrity optimization [3]. A related preprint extended the resource-management formulation and described evaluation with CloudSim and Google Cluster traces [4]. Taken together, these studies provide the foundation for the present work. They also leave an important question open: if workload, anomaly risk and prediction confidence vary over time, should the intensity of integrity validation remain fixed?

We address that question by adding a risk-aware adaptive control layer. Instead of a fixed security cost or a single validation threshold, the controller determines the validation intensity in terms of the current integrity risk, the workload pressure, and the prediction uncertainty. Then the decision is taken into account along with VM consolidation and migration. In practice, the controller might increase validation as risk grows, relax unneeded checks during steady low-risk periods, and deny energy-saving measures if the resulting integrity exposure would be too high.

1.1 Contributions

- A five-term optimization objective that jointly captures energy, residual integrity exposure, SLA penalty, validation overhead and migration cost.
- An adaptive integrity validation function based on anomaly risk and workload prediction uncertainty, extending the fixed-threshold validation used in earlier integrated designs.
- A migration gate that disallows migration actions with predicted integrity and migration penalties larger than the predicted energy benefits.
- **A 30-run reproducible evaluation over 720 hourly intervals per run, with paired inputs, 95% confidence intervals, and paired statistical testing across four policies.**
- **Ablation and sensitivity studies that isolate the value of prediction uncertainty and migration gating and quantify the energy-integrity effect of validation coefficients.**
- A cloud implementation map that connects telemetry, anomaly intelligence, adaptive validation, objective-based candidate ranking, auto-scaling, and consolidation decisions to practical cloud services.

2. Related Work

Dynamic VM consolidation is a well-established approach to energy-aware cloud management. Beloglazov and Buyya showed that online consolidation can lower energy consumption while maintaining service quality [5]. Later studies have explored the same trade-off using multi-resource models, adaptive thresholds, clustering, game-theoretic methods and SLA-aware placement [6-9]. Much of this literature, however, optimizes energy, QoS or both, while security and integrity are usually handled outside the placement decision.

Security-aware VM management brings its own resource costs. Techniques such as secure migration, selective encryption, attestation and integrity checking can reduce risk, but they also consume computation and communication capacity. Our earlier review highlighted this separation between energy and security objectives [1]. AegisCloud contributed a stronger source of security intelligence through robust log analysis [2], and the integrated 2026 architecture subsequently introduced integrity risk into predictive resource consolidation [3]. That study reported a 20-25% energy reduction, SLA compliance in the 93-97% range and stronger integrity protection in its AWS and trace-driven evaluation. The SSRN study [4] further formalized the multi-objective resource-management problem and reported roughly 20-30% energy reduction in its stated evaluation.

The issue examined here is narrower: how the security control itself should change over time. Recent reviews of AI-based cloud resource allocation show growing use of prediction and learning, while also identifying open questions around sustainability and evaluation [10]. Location- and carbon-aware placement expands the objective space further [11]. We keep the present study focused on energy and integrity, but design the controller so that a time-varying signal such as carbon intensity could be added later.

Table 1. Evolution of the authors' research and motivation for the proposed extension.

Research stage	Primary focus	Limitation addressed by next stage
Paper 1 [1]	Review of energy efficiency and cloud security mechanisms	Energy and security mostly treated separately
Paper 2 [2]	Robust cloud log anomaly analysis	Security intelligence not directly coupled to resource control
Paper 3 [3]	Integrated energy-integrity architecture with predictive consolidation	Validation policy remains comparatively coarse
Paper 4 [4]	Multi-objective energy-efficient and integrity-aware resource management	Dynamic validation intensity and uncertainty coupling can be strengthened
This paper	Adaptive validation, uncertainty-aware control, migration gating	Extends the control policy without repeating the earlier architecture

3. Proposed Work

3.1 Design Requirements and Optimization Goal

The proposed controller is designed around four requirements. First, every VM must be placed on exactly one active host without violating CPU or memory capacity. Second, integrity protection must have a non-zero floor and must increase when security evidence or forecast uncertainty increases. Third, consolidation must not be accepted when its expected energy benefit is smaller than the combined migration, SLA, and integrity penalty. Fourth, the controller must avoid oscillation caused by small variations in workload or risk. The optimization goal is therefore not minimum energy alone. It is the minimum feasible combined cost over energy, residual integrity exposure, SLA stress, validation overhead, and migration cost at each control interval.

3.2 System Model

We consider a cloud data centre with H physical hosts and a set of virtualized workloads that changes over time. At control interval t , $W(t)$ denotes workload demand after normalization to the available capacity. The controller receives host-utilization measurements, SLA indicators, integrity alerts and a workload prediction. The risk estimator returns $R(t)$ in the range $[0,1]$; larger values mean that the observed evidence is more consistent with integrity-related abnormal activity. We also use $U(t)$ in $[0,1]$ to describe prediction uncertainty, based on the mismatch between expected and observed demand.

At each interval, the controller decides which hosts remain active, where VMs are placed, whether a migration should occur and how strongly integrity should be validated. These decisions are represented by $A(t)$, $X(t)$, $M(t)$ and $V(t)$, respectively. A value of $V(t)=0$ corresponds to the minimum configured validation effort, whereas $V(t)=1$ represents the highest configured level. The important difference from a fixed policy is that $V(t)$ can be revised at every control interval.

3.2.1 System Entities, State Variables and Decision Variables

The data centre is represented by the host set $H=\{h_1, \dots, h_H\}$ and the virtual-machine set $N(t)=\{n_1, \dots, n_{N(t)}\}$. Each host h has CPU capacity C_h , memory capacity M_h and an active-state indicator $A_h(t)$. Each VM n has CPU demand $c_n(t)$, memory demand $m_n(t)$, predicted demand $\hat{c}_n(t)$, integrity-risk contribution $r_n(t)$ and placement variable $x_{nh}(t)$. The binary variable $x_{nh}(t)=1$ only when VM n is assigned to host h . A migration variable $m_{nig}(t)$ records whether the placement of a VM changes between two consecutive control intervals. These variables separate the physical state of the cloud from the controller decisions and make the optimization constraints explicit.

The controller state at interval t is $S(t)=\{W(t), R(t), U(t), L(t), A(t-1), X(t-1)\}$, where $W(t)$ is normalized workload pressure, $R(t)$ is the aggregate integrity-risk score, $U(t)$ is workload-prediction uncertainty, $L(t)$ summarizes SLA stress, and the previous active-host and placement states provide migration context. The action is $D(t)=\{A(t), X(t), M(t), V(t)\}$. This state-action definition is used consistently by the validation controller, migration gate and objective function.

Table 2. Symbols and decision variables used in the proposed methodology.

Symbol	Meaning	Range / type
H	Number of physical hosts	positive integer
$N(t)$	Number of active VMs at interval t	positive integer
$x_{nh}(t)$	VM-to-host assignment	binary
$A_h(t)$	Host active state	binary
$W(t)$	Normalized workload pressure	$[0,1]$
$R(t)$	Aggregate integrity-risk score	$[0,1]$
$U(t)$	Prediction uncertainty	$[0,1]$
$V(t)$	Adaptive validation intensity	$[V_{min}, V_{max}]$
$M(t)$	Migration decision set	binary per candidate
$J(t)$	Composite decision cost	non-negative

3.3 Adaptive Validation Function

The validation intensity is computed as:

$$V(t) = \text{clip}[v_0 + k_r R(t) + k_u U(t) + k_w W(t)R(t), V_{min}, V_{max}]$$

In this expression, v_0 is the level of protection baseline. The coefficient k_r controls the degree of reaction of validation to the integrity risk. The coefficient k_u increases the checking when the prediction of the workload is uncertain. The last interaction term is governed by k_w and captures situations in which high workload and high risk co-occur. This is important because the same anomaly score can have very different consequences on a lightly loaded system and on a system that is already heavily consolidated.

3.3.1 Risk and Uncertainty Construction

$R(t)$ is treated as a normalized control signal rather than a breach probability. In an implementation, it can combine anomaly evidence, failed integrity checks, unusual access activity and migration-related alerts. Let $z_i(t)$ denote a normalized security indicator and w_i its non-negative importance weight. A practical aggregation is $R(t)=\text{clip}[(\sum_i w_i z_i(t))/(\sum_i w_i), 0, 1]$. This formulation allows AegisCloud-style anomaly confidence to be included as one input without requiring the resource controller to reproduce the anomaly detector itself.

Prediction uncertainty is computed from recent forecast error. If $\hat{W}(t)$ is the predicted normalized workload and $W(t)$ is the observed workload, the instantaneous absolute error is $e(t) = |W(t) - \hat{W}(t)|$. To avoid reacting to one noisy observation, the controller can use an exponentially weighted estimate $U(t) = \rho U(t-1) + (1-\rho)e(t)$, followed by clipping to $[0,1]$. The smoothing factor ρ therefore controls how quickly uncertainty reacts to forecasting errors. This definition also makes the uncertainty term reproducible and separates it from the workload magnitude itself.

3.3.2 Validation Capacity Overhead

Validation consumes CPU and communication capacity. The effective CPU demand of VM n is therefore modeled as $c_{\text{eff},n}(t) = c_n(t)[1 + \eta V(t)]$, where η is the maximum proportional validation overhead. The controller uses $c_{\text{eff},n}(t)$, rather than raw $c_n(t)$, during feasibility checks. This coupling is important because increasing security effort can otherwise create an unmodeled SLA violation when hosts are already highly utilized.

3.4 Multi-Objective Cost

$$J(t) = aE En(t) + 1458 Ir(t) + aS S(t) + aV Ov(t) + aM Mc(t)$$

Here $En(t)$ denotes normalized energy use, $Ir(t)$ the integrity exposure that remains after validation, $S(t)$ the SLA penalty, $Ov(t)$ the validation overhead and $Mc(t)$ the migration cost. Their relative importance is controlled by the non-negative weights aE , aI , aS , aV and aM . We model residual integrity exposure as $R(t)[1 - V(t)]^q$, with q representing validation effectiveness. Including this term prevents an apparently energy-efficient solution from being achieved simply by reducing or disabling security work.

3.4.1 Optimization Constraints

The optimization is subject to one-placement, host-capacity, activation and service constraints. Every active VM must be assigned to exactly one host: $\sum_h x_{nh}(t) = 1$ for each n . CPU feasibility requires $\sum_n x_{nh}(t) c_{\text{eff},n}(t) \leq A_h(t) C_h$, and memory feasibility requires $\sum_n x_{nh}(t) m_n(t) \leq A_h(t) M_h$. These constraints prevent a mathematically attractive objective value from producing an infeasible placement.

SLA stress is represented by the amount by which predicted host utilization exceeds a safe operating threshold θ . For host h , $sh(t) = \max[0, u_h(t) - \theta] / (1 - \theta)$, and $S(t)$ is the mean or weighted mean of $sh(t)$ across active hosts. Residual integrity exposure is $Ir(t) = R(t)[1 - V(t)]^q$, where $q > 0$ describes validation effectiveness. Validation overhead is $Ov(t) = \eta V(t)$ times the validated workload fraction. Migration cost $Mc(t)$ combines transferred VM state and the number of migrations so that repeated movement is penalized.

For comparability, $En(t)$, $Ir(t)$, $S(t)$, $Ov(t)$ and $Mc(t)$ are normalized before weighting. The five objective weights are non-negative and sum to one. This prevents a metric with a numerically larger scale from dominating only because of its unit. The controller minimizes $J(t)$ over feasible candidate actions, not over infeasible placements that violate CPU, memory or SLA constraints.

In the controlled implementation, candidate active-host counts are generated around the predicted demand using a 5% safety margin. For each candidate, predicted energy, SLA stress, residual integrity exposure, validation overhead, and migration distance are computed on normalized scales. The proposed policy uses weights $aE=0.31$, $aI=0.31$, $aS=0.20$, $aV=0.08$, and $aM=0.10$. These weights sum to one and intentionally give equal primary importance to energy and integrity while reserving explicit penalties for SLA, validation overhead, and migration churn. Baseline policies use their own documented operating rules so that each comparison reflects the intended design philosophy rather than forcing every baseline to use the proposed weight vector.

Table 3. Objective components and their operational interpretation.

Component	Operational meaning	Reason included
$En(t)$	Normalized host energy use	Discourages unnecessary active capacity
$Ir(t)$	Residual integrity exposure after validation	Prevents energy savings from weakening protection
$S(t)$	SLA stress / violation penalty	Protects service quality
$Ov(t)$	Validation computation overhead	Avoids permanently excessive checking
$Mc(t)$	Migration and movement cost	Limits churn and migration-induced risk

3.5 Migration Gate

A proposed consolidation or migration is accepted only if its expected energy saving is greater than the combined cost associated with migration, SLA impact and integrity risk. In simplified form, the condition is $\Delta E > I_m \Delta M + I_s \Delta S + I_i \Delta I$. Unlike a global threshold, this gate considers the expected marginal effect of each candidate migration, making it a direct extension of integrity-aware consolidation.

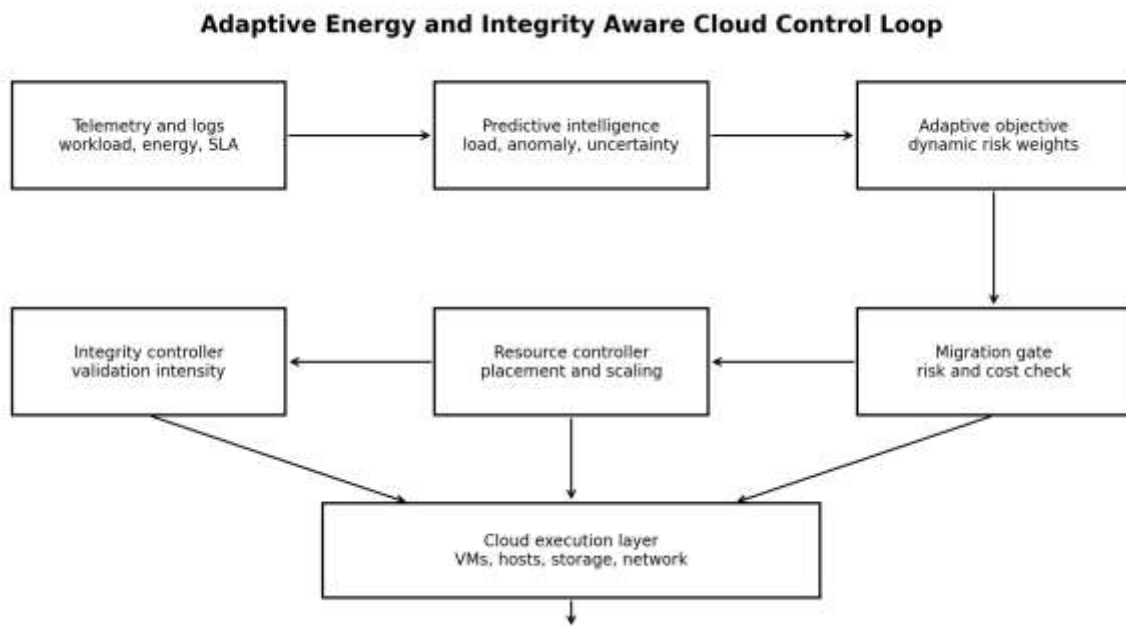


Figure 1(a). Proposed adaptive energy and integrity aware control architecture.

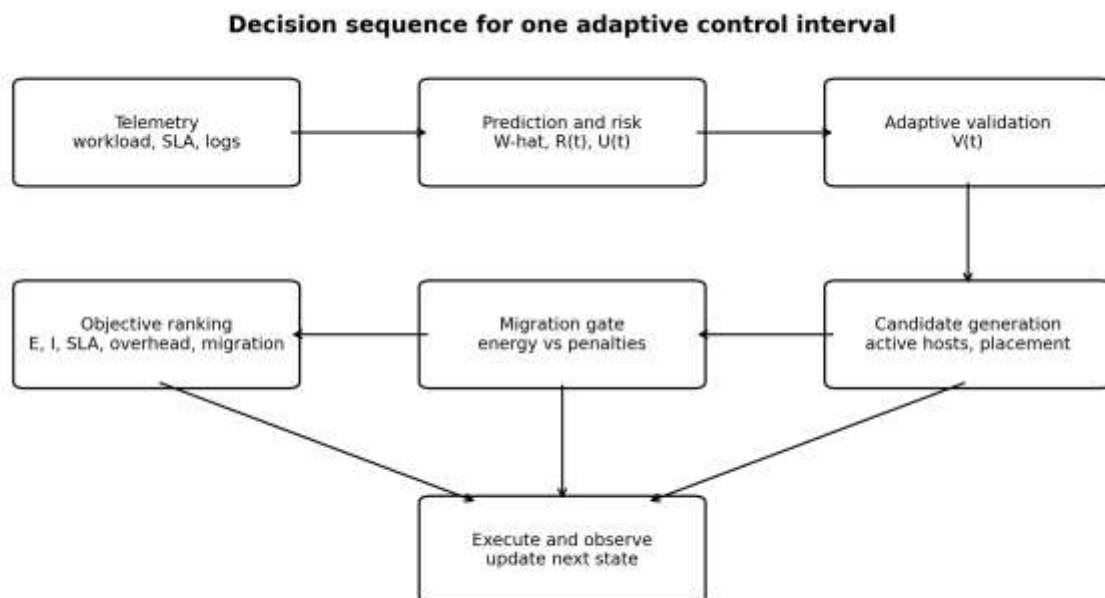


Figure 1(b). Detailed decision sequence for one adaptive control interval.

3.5.1 Candidate Evaluation and Stability Controls

For each candidate migration, ΔE is the predicted reduction in normalized energy during the next decision horizon. ΔM represents migration overhead, ΔS the change in SLA stress and ΔI the change in residual integrity exposure. The candidate is admissible only when $\Delta E > \lambda_m \Delta M + \lambda_s \Delta S + \lambda_i \Delta I$ and all capacity constraints remain satisfied. The λ coefficients translate the three penalties to the same normalized decision scale.

To prevent oscillation, a practical implementation should add three stability controls: an uncertainty-smoothed risk signal, a minimum dwell time before moving the same VM again, and hysteresis between host activation and deactivation thresholds. These controls do not change the optimization objective; they restrict the candidate set so that small measurement fluctuations do not trigger repeated migrations.

3.6 Adaptive Control Algorithm

The controller repeats the same decision cycle at every interval. It first gathers workload, host-utilization, energy, SLA and security measurements. The workload predictor then estimates near-term demand and reports how uncertain that estimate is. In parallel, security-related logs and events are converted into the integrity-risk score $R(t)$. Using these signals, the controller calculates $V(t)$ and accounts for the extra capacity required by validation.

Candidate placement, scaling and consolidation actions are then examined through the migration gate. The feasible action with the lowest overall cost is applied, and the observed outcome becomes input to the next control interval.

Algorithm 1: Adaptive Risk-Aware Energy-Integrity Resource Control

1. Input host capacities, previous placement, current workload and SLA telemetry, workload forecast, security indicators, model coefficients, and objective weights.
2. Normalize workload and security indicators; compute $W(t)$, aggregate $R(t)$, and update smoothed uncertainty $U(t)$.
3. Compute adaptive validation intensity $V(t)$ and effective VM demand including validation overhead.
4. Apply a 5% forecast safety margin and generate feasible candidate active-host counts and VM placements that satisfy CPU and memory limits.
5. For each candidate, estimate normalized energy $En(t)$, SLA stress $S(t)$, residual integrity exposure $Ir(t)$, validation overhead $Ov(t)$, and migration cost $Mc(t)$.
6. For proposed-policy down-scaling actions during high-risk intervals, apply the migration gate and reject candidates whose energy benefit does not exceed the weighted movement, SLA, and integrity penalties.
7. Evaluate $J(t)=aE En(t)+aI Ir(t)+aS S(t)+aV Ov(t)+aM Mc(t)$ for the remaining feasible candidates.
8. Select the candidate with minimum $J(t)$; if all down-scaling candidates are rejected, retain the current host count or perform required scale-up.
9. Execute the chosen placement, scaling, migration, and validation action; record performance and update state for the next interval.

4. Results and Discussion

4.1 Simulation Model and Experimental Methodology

The revised evaluation uses a controlled simulator specifically constructed to test the adaptive controller. It does not reuse the numerical results of the earlier AWS or Google Cluster trace studies [3], [4]. Each run contains 720 one-hour control intervals and 100 normalized hosts. The same workload, forecast, uncertainty, and risk realization is supplied to all compared policies within a run. This paired-input design isolates policy effects while preserving stochastic variation across independent runs.

Thirty independent stochastic runs are executed using seeds 100-129. Workload combines daily and weekly periodicity, Gaussian variation, and six randomly placed burst windows. Integrity risk combines a workload-correlated background term, beta-distributed variation, and four injected high-risk episodes. A one-step workload predictor uses the previous observed load, a periodic component, and random forecast error; prediction uncertainty is the exponentially smoothed absolute error scaled to $[0,1]$. Seed 42 is used only for illustrative time-series figures and is not included in the 30-run statistical summary.

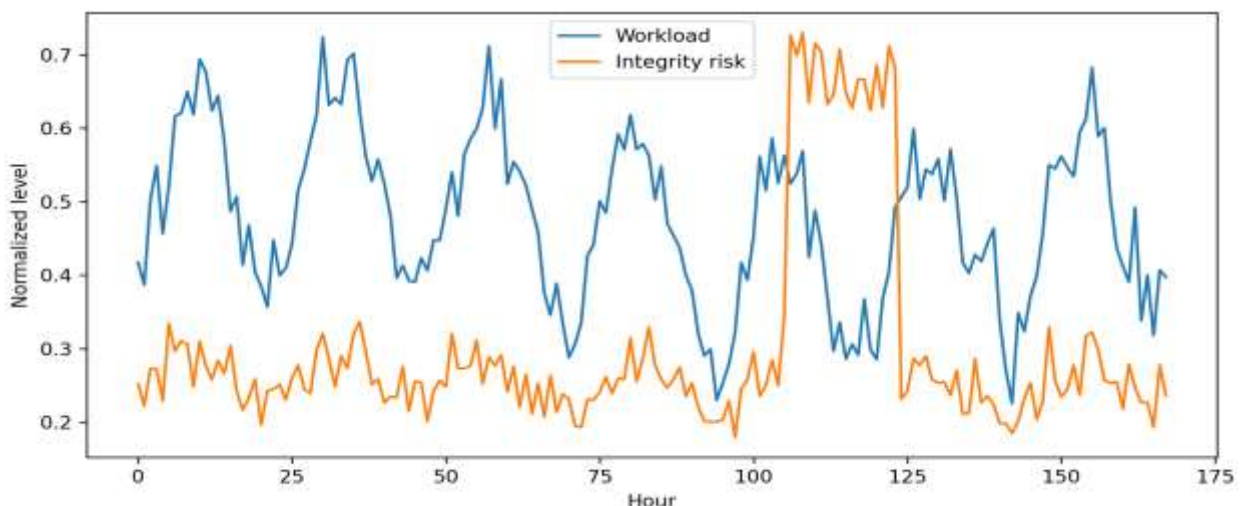


Figure 2. Example first-week workload and integrity risk signals used in the controlled simulation.

4.1.1 Reproducible Workload, Risk and Power Construction

For run r and interval t , workload $W_r(t)$ is generated from daily and weekly periodic components, zero-mean Gaussian variation, and six burst windows. Risk $R_r(t)$ contains a workload-correlated background component, beta-distributed variation, and four injected high-risk episodes. Forecast uncertainty $U_r(t)$ is updated as $U_r(t)=\text{clip}[3.2(0.82U_r(t-1)+0.18|W_r(t)-\hat{W}_r(t)|),0,1]$. All four policies receive exactly the same $W_r(t)$, $R_r(t)$, and $U_r(t)$ sequences within each run.

Host power is modeled by $P_h(t) = A_h(t)[0.52 + 0.48u_h(t)^{1.45}]$, where $u_h(t)$ is normalized CPU utilization. A 5% forecast safety margin is applied when candidate host counts are generated. Total interval energy is the sum across active hosts. Because the analysis reports relative energy rather than watts for a named server model, the power equation is used consistently for every policy and every run.

For the proposed policy, $V(t) = \text{clip}[0.18 + 0.68R(t) + 0.35U(t), 0.18, 0.92]$. The lower bound preserves a protection floor and the upper bound prevents validation from consuming unlimited capacity. The risk coefficient is intentionally larger than the uncertainty coefficient. The revised paper does not treat these values as universal constants: Section 4.2.6 varies both coefficients to test sensitivity.

Table 4. Experimental variables, construction and role in evaluation.

Variable	Construction in this experiment	Purpose
$W(t)$	Daily + weekly demand + Gaussian variation + six bursts	Workload pressure
$R(t)$	Workload-correlated background + beta variation + four risk episodes	Integrity-risk stimulus
$U(t)$	Smoothed workload prediction error	Forecast confidence
$P_h(t)$	$0.52 + 0.48u^{1.45}$ for active host	Relative host power
$V(t)$	$\text{clip}[0.18 + 0.68R(t) + 0.35U(t), 0.18, 0.92]$	Adaptive validation effort
Seed	30 independent seeds (100-129); seed 42 only for figures	Robustness and repeatability

4.1.2 Compared Policies

Table 5. Comparison of resource and validation behavior of the evaluated policies.

Policy	Resource behavior	Validation behavior
Energy only	Aggressive consolidation	Low fixed validation
Fixed validation	Conservative headroom	High constant validation
Prior joint policy	Risk-aware resource management	Risk-proportional validation
Proposed adaptive policy	Risk-aware consolidation with migration gate	Risk plus uncertainty adaptive validation

4.1.3 Evaluation Metrics

We compare the policies using relative energy consumption, SLA violation rate, residual integrity exposure, validation overhead, migration index, average number of active hosts and mean CPU utilization. Energy values are normalized against the energy-only policy. The integrity-exposure value should be read as a relative decision index, not as the probability of a successful compromise. The migration index is likewise normalized and is intended only for comparisons within this controlled experiment.

4.1.4 Metric Computation and Fair Comparison

Relative energy is computed as 100 times the cumulative energy of a policy divided by cumulative energy of the energy-only baseline. SLA violation rate is the percentage of control intervals in which the configured service threshold is exceeded. Integrity exposure is the cumulative or mean residual index $R(t)[1-V(t)]^q$ under the same risk sequence. Validation overhead is the validation-related compute term expressed as a percentage of available capacity. The migration index is a normalized movement-cost measure used only for within-experiment comparison. Mean active hosts and CPU utilization describe consolidation behavior.

Within each seed, all policies are evaluated over identical 720-interval workload and risk inputs. Results are summarized across 30 independent seeds as mean values with 95% confidence intervals calculated as $1.96s/\sqrt{30}$. Because policies are evaluated on the same stochastic realization, paired two-sided t-tests are used for selected proposed-versus-baseline comparisons. The purpose of the tests is to assess repeatability of the controlled-simulation differences, not to claim population-level performance for real data centres.

4.1.5 Statistical, Ablation and Sensitivity Protocol

Three robustness checks are added. First, paired statistical tests compare the proposed controller with the prior joint and fixed-validation policies over the same 30 seeds. Second, an ablation study removes prediction-uncertainty coupling and then removes the migration gate to isolate their incremental effects. Third, the risk coefficient k_r and uncertainty coefficient k_u are varied over a 4×4 grid using ten paired seeds. These analyses use the same workload generator and power model as the primary experiment.

4.2 Results

Table 6. Thirty-run performance summary (mean $\pm$ 95% confidence interval).

Policy	Energy %	SLA viol. %	Integrity exposure	Validation ovh. %	Migration index
Energy only	100.00 $\pm$ 0.00	14.07 $\pm$ 0.34	21.96 $\pm$ 0.11	0.18 $\pm$ 0.00	6.10 $\pm$ 0.06
Fixed validation	101.69 $\pm$ 0.00	13.88 $\pm$ 0.35	3.66 $\pm$ 0.02	1.02 $\pm$ 0.00	6.19 $\pm$ 0.06
Prior joint	101.02 $\pm$ 0.00	13.81 $\pm$ 0.37	8.54 $\pm$ 0.02	0.68 $\pm$ 0.00	6.11 $\pm$ 0.06
Proposed adaptive	101.34 $\pm$ 0.04	13.55 $\pm$ 0.35	6.96 $\pm$ 0.02	0.77 $\pm$ 0.00	6.01 $\pm$ 0.06

Table 7. Resource-utilization behavior across 30 runs (mean $\pm$ 95% confidence interval).

Policy	Active hosts	CPU utilization %
Energy only	55.69 $\pm$ 0.13	91.03 $\pm$ 0.07
Fixed validation	56.65 $\pm$ 0.13	90.99 $\pm$ 0.07
Prior joint	56.28 $\pm$ 0.13	90.98 $\pm$ 0.07
Proposed adaptive	56.58 $\pm$ 0.14	90.71 $\pm$ 0.09

Table 6 shows that the proposed adaptive policy occupies a middle operating region rather than dominating every metric. Across 30 runs it uses 101.34% relative energy, compared with 101.02% for the prior joint policy and 101.69% for fixed validation. Its mean integrity-exposure index is 6.96, lower than the prior joint policy (8.54) but higher than fixed validation (3.66). The proposed policy records 13.55% SLA violations, 0.77% validation overhead, and a migration index of 6.01. Table 7 shows that it maintains 56.58 active hosts on average with 90.71% mean CPU utilization. These values are consistent with a controller that deliberately spends some additional protection effort while avoiding the permanent overhead of fixed high validation.

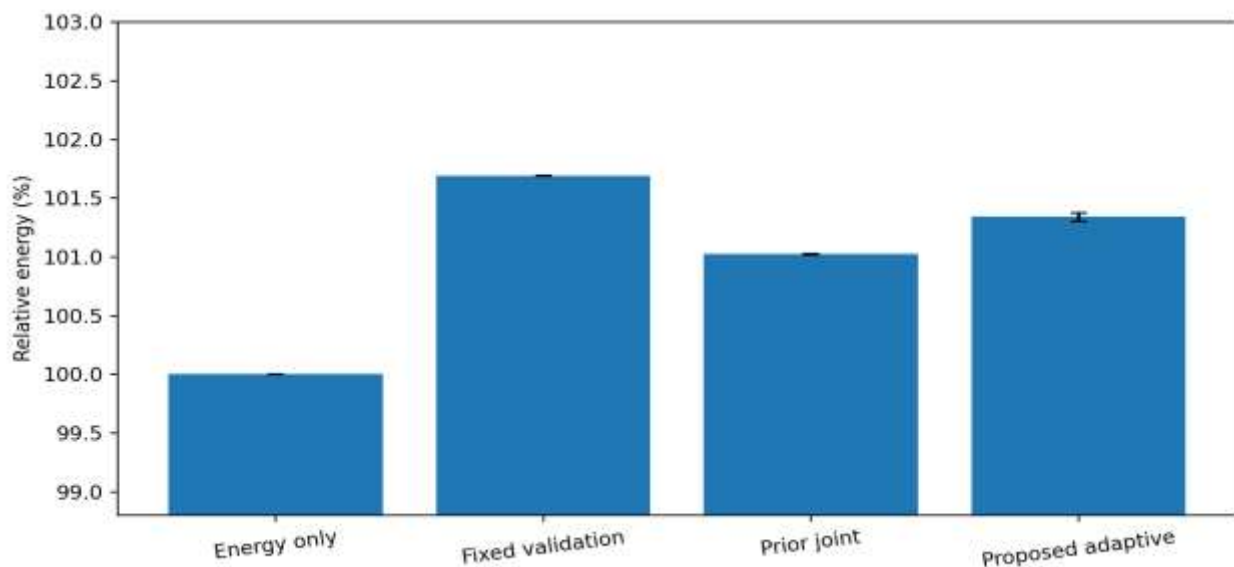


Figure 3. Mean relative energy across 30 independent runs with 95% confidence intervals.

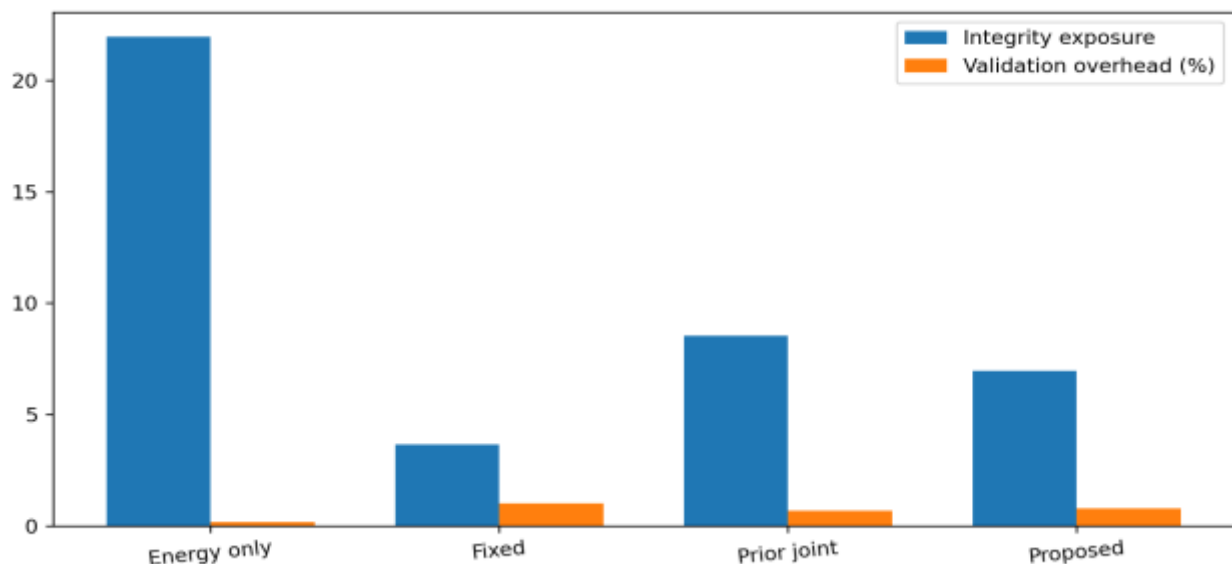


Figure 4. Mean residual integrity exposure and validation overhead across 30 independent runs.

4.2.1 Adaptive Behavior Over Time

The policies' behavior over time is illustrated in Figure 5. Fixed validation remains constant. The prior joint policy changes mainly with the risk signal, whereas the proposed controller also responds to prediction uncertainty. Validation therefore rises around workload spikes or abnormal-risk periods and returns toward the protection floor as conditions stabilize. In this way, validation behaves as a controllable resource load rather than a permanently fixed background task.

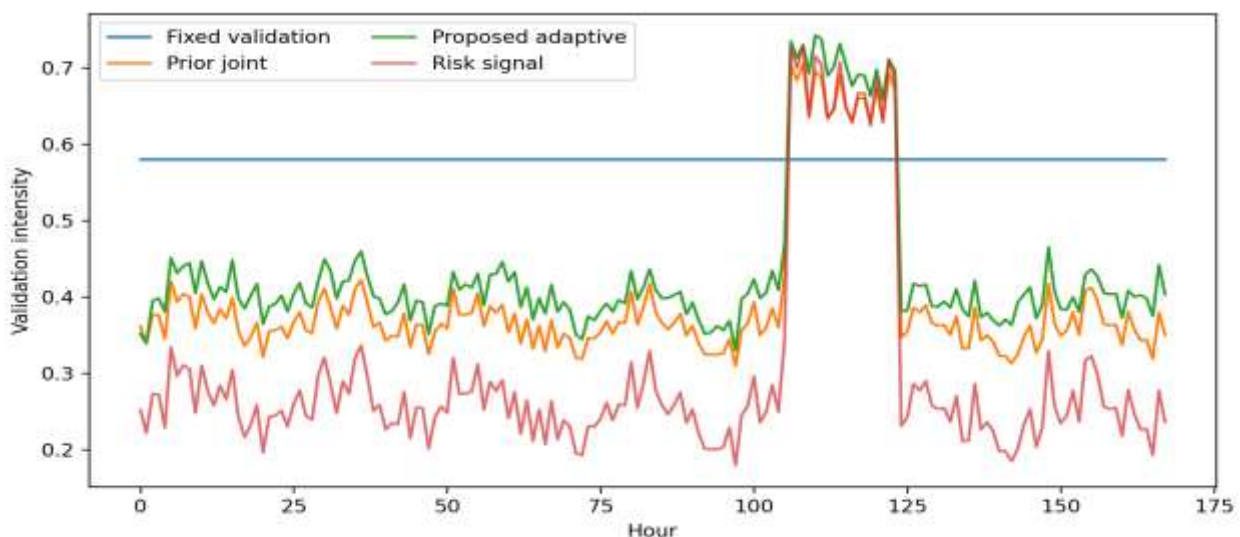


Figure 5. Validation intensity over the first week of representative seed 42.

4.2.2 Energy and Integrity Trade-off

Energy alone does not give a complete picture of policy quality. Figure 6 plots the 30-run mean relative energy against mean residual integrity exposure. The energy-only policy has the lowest energy but the highest exposure. Fixed validation has the lowest exposure but the highest energy and validation overhead. The prior joint and proposed adaptive policies lie between these extremes. Relative to the prior joint policy, the proposed controller moves toward lower integrity exposure at a modest energy cost; relative to fixed validation, it moves toward lower energy and lower validation overhead while accepting more residual exposure. This is the intended multi-objective trade-off rather than a claim of universal Pareto optimality.

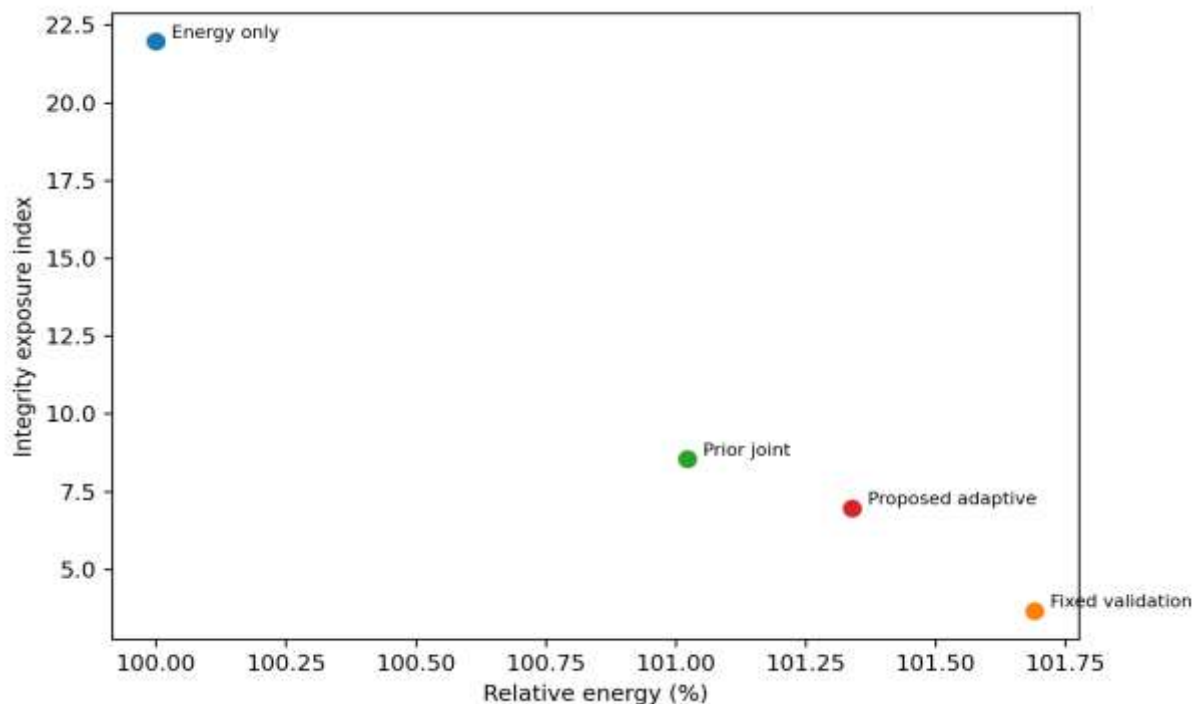


Figure 6. Mean energy-integrity operating points across 30 independent runs.

4.2.3 Relation to Earlier Experimental Evidence

The controlled experiment in this paper should be read alongside the earlier studies, but not merged with them. The integrated architecture in [3] reported a 20-25% reduction in energy use and SLA compliance of about 93-97% in its AWS and workload-trace evaluation. The SSRN manuscript [4] described an approximately 20-30% energy reduction using CloudSim and Google trace data. Those earlier results provide evidence that joint energy-integrity management is feasible. The present simulation serves a different purpose: it isolates the additional effect of time-varying validation and migration gating under a transparent, repeatable workload.

4.2.4 Mechanistic Interpretation of the Observed Results

The numerical pattern in Tables 6 and 7 is consistent with the controller structure. Energy-only control keeps validation near its floor and therefore minimizes protection overhead, but it leaves the largest residual integrity exposure. Fixed validation applies $V=0.58$ throughout the experiment and therefore provides the strongest protection in this model at the cost of the highest validation overhead and energy. The prior joint policy varies validation with risk. The proposed policy additionally uses uncertainty and a risk-aware migration gate, producing lower exposure and fewer migrations than the prior joint policy while consuming slightly more energy.

Across the 30 runs, the proposed policy reduces integrity exposure from 8.54 to 6.96 relative to the prior joint policy, an 18.4% reduction. Its SLA-violation rate decreases from 13.81% to 13.55%, and its migration index decreases from 6.11 to 6.01. These gains are accompanied by an increase in relative energy from 101.02% to 101.34% and validation overhead from 0.68% to 0.77%. Compared with fixed validation, the proposed controller reduces relative energy by 0.35 percentage points and validation overhead by 0.25 percentage points but has higher residual exposure. This pattern demonstrates that the adaptive mechanism changes where the system operates on the energy-protection trade-off rather than merely shifting all metrics in the same direction.

These comparisons are causal only within the controlled simulator because all policies receive paired inputs and differ only in decision logic. They should not be generalized to heterogeneous production servers or real attack distributions without external validation. The added multi-seed, statistical, ablation, and sensitivity analyses strengthen internal validity and repeatability, while external validity remains a limitation discussed in Section 4.4.

4.2.5 Statistical Repeatability Across 30 Runs

Paired two-sided t-tests were applied because every policy was evaluated on the same workload and risk realization for each seed. Compared with the prior joint policy, the proposed controller shows lower integrity exposure (mean paired difference -1.57, $p<0.001$), lower SLA-violation rate (-0.26 percentage points, $p<0.001$), and a lower migration index (-0.10, $p<0.001$), while relative energy increases by 0.32 percentage points ($p<0.001$). Compared with fixed validation, the proposed policy uses 0.35 percentage points less relative energy and 0.25 percentage points less validation overhead (both $p<0.001$), but its integrity exposure is higher. The statistical results therefore reinforce the trade-off interpretation rather than supporting a claim that one policy is best on every objective.

Table 8. Selected paired comparisons across 30 runs.

Comparison	Metric	Mean difference	p-value
Proposed - Prior joint	Energy (%)	+0.32	<0.001
Proposed - Prior joint	SLA violation (%)	-0.26	<0.001
Proposed - Prior joint	Integrity exposure	-1.57	<0.001
Proposed - Prior joint	Migration index	-0.10	<0.001

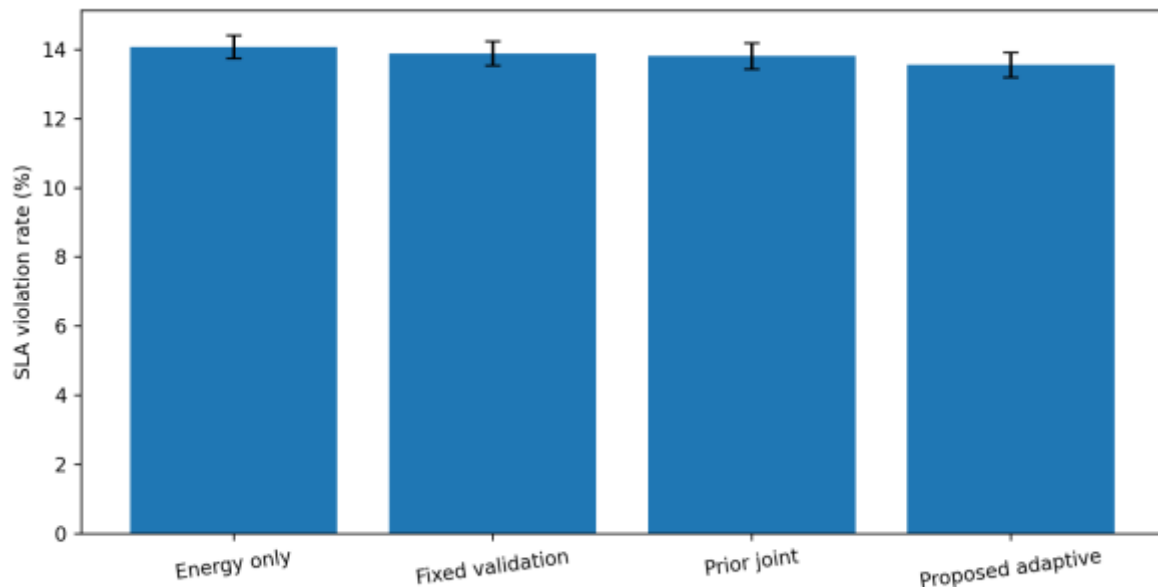


Figure 7. Mean SLA-violation rate across 30 independent runs with 95% confidence intervals.

4.2.6 Ablation Study

The ablation study separates the contribution of uncertainty coupling and migration gating. Removing $U(t)$ from the validation function reduces relative energy slightly to 101.02% but increases integrity exposure from 6.96 to 8.92. Removing the migration gate leaves integrity exposure unchanged because the validation rule is unchanged, but increases the migration index from 6.01 to 6.09. These results show that uncertainty primarily influences protection intensity, whereas the migration gate primarily limits movement during high-risk conditions.

Table 9. Ablation results across 30 runs (mean $\pm$ 95% confidence interval).

Variant	Energy %	SLA viol. %	Integrity exposure	Val. ovh. %	Migration index
Proposed adaptive	101.34 $\pm$ 0.04	13.55 $\pm$ 0.35	6.96 $\pm$ 0.02	0.77 $\pm$ 0.00	6.01 $\pm$ 0.06
No uncertainty	101.02 $\pm$ 0.01	13.68 $\pm$ 0.37	8.92 $\pm$ 0.02	0.66 $\pm$ 0.00	6.07 $\pm$ 0.06
No migration gate	101.24 $\pm$ 0.01	13.76 $\pm$ 0.36	6.96 $\pm$ 0.02	0.77 $\pm$ 0.00	6.09 $\pm$ 0.06
Fixed validation	101.69 $\pm$ 0.00	13.88 $\pm$ 0.35	3.66 $\pm$ 0.02	1.02 $\pm$ 0.00	6.19 $\pm$ 0.06

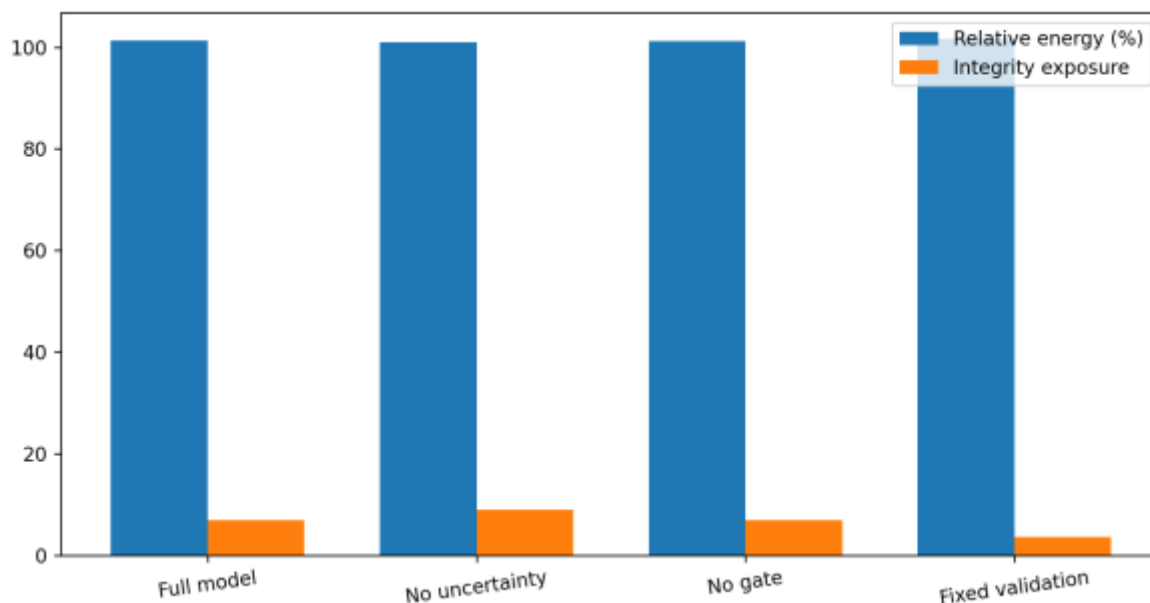


Figure 8. Ablation comparison showing the effect of uncertainty coupling and migration gating.

4.2.7 Sensitivity to Validation Coefficients

Sensitivity analysis varies k_r in $\{0.50, 0.60, 0.68, 0.76\}$ and k_u in $\{0.20, 0.28, 0.35, 0.42\}$ over ten paired seeds. Increasing either coefficient strengthens validation and monotonically reduces residual integrity exposure, while relative energy and validation overhead rise gradually. Over the tested grid, relative energy remains between approximately 101.12% and 101.47%, whereas integrity exposure decreases from 9.52 to 5.99. The selected setting $k_r=0.68$ and $k_u=0.35$ lies inside this range rather than at an extreme, supporting its use as a balanced operating point for the controlled study.

Table 10. Representative coefficient-sensitivity settings (10-run means).

k_r	k_u	Energy %	SLA viol. %	Integrity exposure	Val. ovh. %
0.50	0.20	101.12	13.46	9.52	0.63
0.60	0.35	101.29	13.61	7.66	0.73
0.68	0.35	101.36	13.58	6.97	0.77
0.76	0.42	101.47	13.58	5.99	0.83

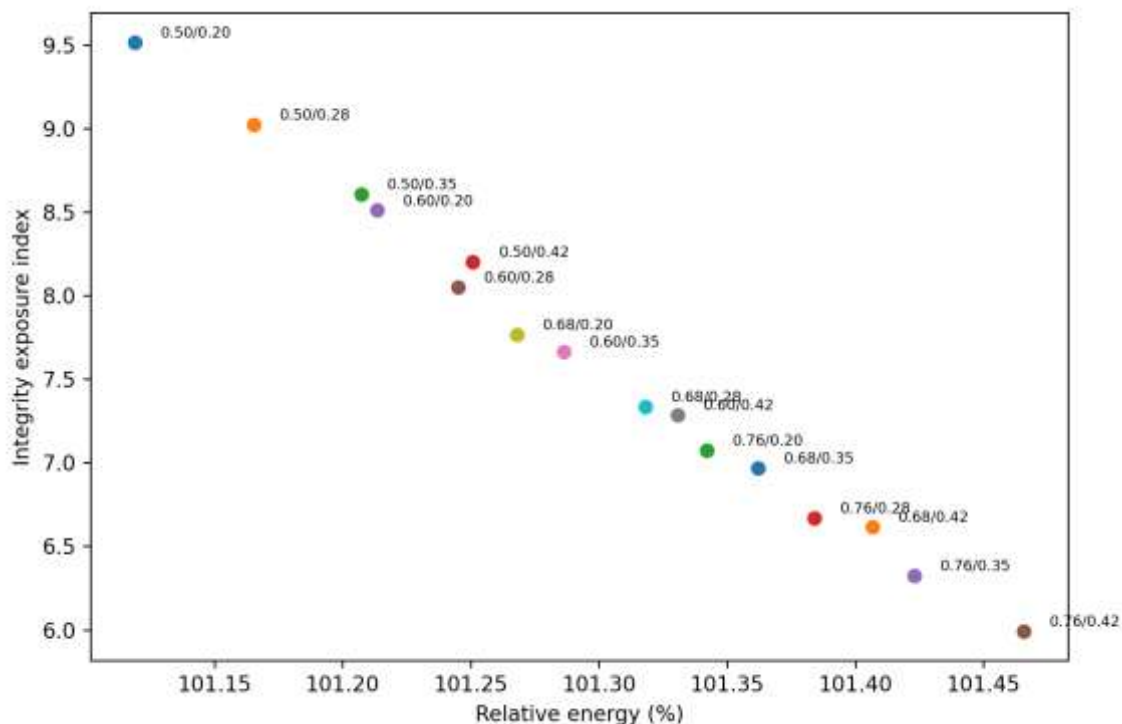


Figure 9. Sensitivity of the energy-integrity operating point to k_r/k_u coefficient pairs.

4.3 Discussion

4.3.1 Why Adaptive Validation Matters

Elasticity in cloud systems is usually discussed in terms of computing capacity, yet protection effort can also be adjusted. When the risk level is low and the workload prediction is reliable, repeatedly running expensive integrity operations at maximum intensity may offer little additional value. The opposite response is also problematic: validation should not be weakened simply to preserve an energy target when risk is increasing. The adaptive approach used here tries to handle both situations by making the amount of checking depend on the available evidence.

4.3.2 Connection with AegisCloud

AegisCloud [2] is useful in this context because an adaptive controller needs a meaningful risk signal. Its log-anomaly output could contribute to $R(t)$, together with conventional integrity checks, attestation failures, unusual access behaviour and migration-related events. We do not retrain or re-evaluate AegisCloud in the present experiment. Instead, this paper identifies the point at which such security intelligence can enter the control loop and influence resource-management decisions.

4.3.3 Practical Cloud Mapping

One practical implementation can be built from familiar AWS services. CloudWatch provides information on utilization and operational performance, Lambda carries out the event-driven control logic, Auto Scaling groups manage scaling actions, and SageMaker is used to host predictive models. IAM or Cognito can be used to enforce identity and access policies, while KMS and the mechanisms for ensuring data integrity protect the keys and the data. The adaptive controller does not have to operate at the same frequency as the raw monitoring; a less frequent resource-management interval can reduce the control overhead. Explicit migration gating is especially relevant in private-cloud, hybrid-cloud and orchestrated virtualization environments where placement and migration decisions are directly available to the controller.

4.3.4 Complexity

A straightforward placement search can become costly when N candidate VMs are considered across H possible hosts. The proposed extension does not depend on exhaustive search. Risk scoring and validation-intensity calculation scale approximately as $O(N)$ with the monitored workloads, while a direct best-fit-style placement step is about $O(NH)$. In practice, the search space can be reduced before placement by removing candidates that fail capacity, trust-level or risk-threshold requirements.

4.4 Threats to Validity and Limitations

The controlled simulator deliberately simplifies several aspects of a production cloud. Hosts are homogeneous and power is normalized, risk events are synthetic, and VM migration is represented by an aggregate movement term rather than packet-level or storage-level transfer. The 30-run analysis reduces dependence on a single random realization but does not convert the experiment into a production benchmark. Numerical values should therefore be interpreted as internally controlled comparative results.

The integrity-exposure index is a control metric, not a measured probability of compromise. Its calibration would need to reflect the application domain and the reliability of the upstream anomaly and integrity signals. The workload predictor is intentionally lightweight so that the study isolates the controller rather than prediction-model complexity. A real deployment should additionally evaluate heterogeneous hosts, real workload traces, measured migration time, network contention, and calibrated security events in CloudSim Plus, OpenStack, or an AWS testbed.

4.5 Reproducibility and Validation Roadmap

The controlled experiment is reproducible from the reported 720-interval horizon, 100-host pool, seed set 100-129, workload and risk construction, forecast rule, uncertainty smoothing, host-power equation, 5% prediction margin, adaptive-validation coefficients, objective weights, and migration-gate rule. The same generator and controller code was used for the main, ablation, and sensitivity analyses; seed 42 is reserved for illustrative plots only.

The revised manuscript now includes the previously missing multi-seed statistical analysis, ablation experiments, and coefficient-sensitivity analysis. External validation remains future work and should preserve the controller while replacing synthetic workload/risk inputs with public workload traces and independently generated or measured integrity events. This separation prevents the controlled results from being misrepresented as live-cloud measurements.

5. Conclusion

This study extends earlier work on secure and energy-efficient cloud management by treating integrity validation as a time-varying control variable rather than a fixed background process. The revised framework formally defines state variables, decision variables, feasibility constraints, risk and uncertainty construction, validation overhead, a five-term normalized objective, migration gating, and stability controls. The implementation evaluates feasible host-count candidates at every interval and selects the minimum-cost action after risk-aware migration screening.

Across 30 independent controlled runs, the proposed policy achieved 101.34% relative energy, 13.55% SLA violations, 6.96 integrity exposure, 0.77% validation overhead, and a migration index of 6.01. Relative to the prior joint policy, integrity exposure decreased by 18.4% with a 0.32 percentage-point increase in relative energy. Ablation showed that uncertainty coupling materially reduced integrity exposure, while migration gating reduced movement during high-risk periods. Sensitivity analysis showed a smooth, interpretable energy-protection trade-off over the tested coefficient grid. These results strengthen the scientific evidence for the adaptive control mechanism while also showing that fixed high validation remains preferable when maximum protection is valued above energy and overhead. External validation with real workload traces and a cloud or simulator testbed is the principal next step.

Data and Code Availability

Our simulation employs workload and risk data generated artificially using a fixed seed for the random number generator. The experimental methodology also provides the necessary information about the generation functions and parameters and how we evaluated the experiment. In our earlier research, we have used Google Cluster workload traces that were made publicly available by Google in ClusterData2019 dataset via BigQuery [12].

Acknowledgement

This study uses a controlled computer simulation and does not involve human participants or animals.

References

- [1] S. Patel and A. Nayak, 'A Comprehensive Analysis of Protocols and Models for Improving Energy Efficiency and Security in Cloud Computing,' 2023 Global Conference on Information Technologies and Communications (GCITC), 2023, doi: 10.1109/GCITC60406.2023.10426325.
- [2] S. H. Patel and A. Nayak, 'AegisCloud: Adversarial Contrastive Transformerbased Framework for Secure Cloud Log Analysis,' 2025 6th International Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 1238-1244, 2025, doi: 10.1109/ICESC65114.2025.11212308.
- [3] S. H. Patel and A. Nayak, 'An Integrated Secure and Energy-Efficient Cloud Architecture using Resource Consolidation and Machine Learning-Based Optimization,' SSRG International Journal of Electronics and Communication Engineering, vol. 13, no. 5, pp. 109-128, 2026, doi: 10.14445/23488549/IJECE-V13I5P111.
- [4] S. Patel and A. Nayak, 'A Multi-Objective Energy-Efficient and Integrity-Aware Resource Management Framework for Cloud Computing,' SSRN, posted July 4, 2026, doi: 10.2139/ssrn.7052117.
- [5] 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, 2012.
- [6] S. Banerjee, S. Roy, and S. Khatua, 'Towards energy and QoS aware dynamic VM consolidation in a multi-resource cloud,' Future Generation Computer Systems, vol. 157, pp. 376-391, 2024, doi: 10.1016/j.future.2024.03.058.

- [7] S. Manikandan et al., 'Efficient energy consumption in hybrid cloud environment using adaptive backtracking virtual machine consolidation,' *Scientific Reports*, vol. 14, 22869, 2024, doi: 10.1038/s41598-024-72459-z.
- [8] V. Kumar et al., 'E2SVM: Electricity-Efficient SLA-aware Virtual Machine Consolidation approach in cloud data centers,' *PLOS ONE*, vol. 19, no. 6, e0303313, 2024, doi: 10.1371/journal.pone.0303313.
- [9] A. Tandon and S. Patel, 'DBSCAN based approach for energy efficient VM placement using medium level CPU utilization,' *Sustainable Computing: Informatics and Systems*, vol. 43, 101025, 2024, doi: 10.1016/j.suscom.2024.101025.
- [10] S. Alam et al., 'AI-driven resource allocation in cloud computing: a systematic review revealing critical sustainability and evaluation gaps,' *Computing*, vol. 108, article 67, 2026, doi: 10.1007/s00607-026-01655-8.
- [11] S. Rawas, A. Zekri, and A. El-Zaart, 'LECC: Location, Energy, Carbon and Cost-Aware VM Placement Model in Geo-Distributed DCs,' *Sustainable Computing: Informatics and Systems*, vol. 33, 2022.
- [12] Google, 'ClusterData2019: Borg cluster workload traces,' Google Cluster Data repository, 2019. Available through Google BigQuery.
- [13] R. Buyya et al., 'Energy-efficiency and sustainability in new generation cloud computing: A vision and directions for integrated management of data centre resources and workloads,' *Software: Practice and Experience*, 2024, doi: 10.1002/spe.3248.
- [14] B. Hu, Y. Shi, G. Chen, Z. Cao, and M. Zhou, 'Workload-Aware Scheduling of Real-Time Jobs in Cloud Computing to Minimize Energy Consumption,' 2024.
- [15] P. Goswami et al., 'Investigation on Storage Level Data Integrity Strategies in Cloud Computing: Classification, Security Obstructions, Challenges and Vulnerability,' *Journal of Cloud Computing*, vol. 13, 2024.
- [16] R. M. Haris et al., 'Enhancing Security and Performance in Live VM Migration: A Machine Learning-Driven Framework with Selective Encryption,' *Expert Systems*, vol. 42, no. 2, 2025.
- [17] Y. S. Patel, R. Jaiswal, and R. Misra, 'Deep Learning-Based Multivariate Resource Utilization Prediction for Hotspots and Coldspots Mitigation in Green Cloud Data Centers,' *The Journal of Supercomputing*, vol. 78, pp. 5806-5855, 2022.
- [18] K. Haghshenas et al., 'MAGNETIC: Multi-Agent Machine Learning-Based Approach for Energy Efficient Dynamic Consolidation in Data Centers,' *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 30-44, 2022.
- [19] G. Verma et al., 'Secure VM Migration in Cloud: Multi-Criteria Perspective with Improved Optimization Model,' *Wireless Personal Communications*, vol. 124, pp. 75-102, 2022.
- [20] A. Alouran et al., 'Energy Efficient Virtual Machines Placement in Cloud Datacenters using Genetic Algorithm and Adaptive Thresholds,' *PLOS ONE*, vol. 19, no. 1, 2024.
- [21] Z. Khodaverdian et al., 'An Energy Aware Resource Allocation based on Combination of CNN and GRU for Virtual Machine Selection,' *Multimedia Tools and Applications*, vol. 83, pp. 25769-25796, 2024.
- [22] F. J. Maldonado-Carrascosa et al., 'Game Theory-Based Virtual Machine Migration for Energy Sustainability in Cloud Data Centers,' *Applied Energy*, vol. 372, 2024.
- [23] S. Parthasarathy, 'OSVR: An Efficient Support Vector Regression Model-Based Host Overload Detection and Secure Virtual Machine Migration,' *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, pp. 7309-7317, 2023.
- [24] N. K. Pandey et al., 'Energy Efficiency Strategy for Big Data in Cloud Environment using Deep Reinforcement Learning,' *Mobile Information Systems*, 2022.
- [25] S. S. Panwar et al., 'Machine Learning Approaches for Efficient Energy Utilization in Cloud Data Centers,' *Procedia Computer Science*, vol. 235, pp. 1782-1792, 2024.