International Journal of Research in Engineering and Management Sciences (IJREMS)
E-ISSN: 3107-7439
Research Article
VOL.: 2, Issue: 6, June 2026

XAI-Driven Green Cloud Intelligence for Transparent and Optimized Resource Allocation in Cloud Computing

[*1]Dr. KISHOR KUMAR GAJULA, [2]Dr. B. SRINIVASA RAO

[1]Associate Professor, Department of CSE, Mother Theressa College Of Engineering &Technology, Peddapalli.https://orcid.org/0009-0003-8141-3332 , E-Mail: drkishorkumarg@gmail.com

[2]Assistant Professor, Department of CSE-Data science, Gurunanak Institutions Technical Campus, Hyderabad.E-mail ID: bsrao.cnu26@gmail.com

ABSTRACT: The energy footprint of hyperscale cloud infrastructure has become a first-order engineering and environmental concern, yet the machine-learning controllers that increasingly govern resource allocation remain opaque, which discourages their adoption by operators who must justify every consolidation or migration decision. This paper introduces *Green Cloud Intelligence* (GCI), an explainable framework that couples an optimised gradient-boosted decision engine with a Shapley-value transparency layer and an energy-aware allocation policy, so that each placement decision is both accurate and accountable. The framework ingests multivariate host telemetry, applies a hybrid feature-selection stage that combines mutual information with SHAP-gain ranking, and trains a class-balanced, hyperparameter-tuned XGBoost classifier that maps each host-hour to one of three operational decisions—consolidate, maintain, or relieve. A physically grounded power and demand model is used to generate a reproducible twelve-feature workload corpus of 12,000 host-hour records against which the framework and seven representative baselines are evaluated. The proposed model attains 95.06% accuracy, a macro-F1 of 0.9387, and a macro one-vs-rest ROC-AUC of 0.9947, outperforming logistic regression, decision tree, *k*-NN, SVM, random forest, and a multilayer perceptron while retaining sub-millisecond inference. When its decisions are translated into a consolidation policy, GCI realises a net data-centre energy reduction of 19.45%, capturing roughly 92% of an oracle upper bound. SHAP attribution shows the decisions are driven by physically meaningful signals—processor utilisation, virtual-machine density, and inlet temperature—rather than spurious correlations, and an ablation study confirms that a compact eight-feature model preserves accuracy at reduced cost. The novelty lies in unifying predictive accuracy, native transparency, and quantified energy accountability within a single deployable pipeline. Future work will validate the framework on public production traces and integrate carbon-intensity forecasting for temporally shifted scheduling.

Keyword: Explainable artificial intelligence, green computing, cloud resource allocation, energy efficiency, SHAP, gradient boosting, virtual machine consolidation, sustainable data centres.

Received Date: 7 June 2026**; Accepted Date:** 17 June 2026**; Published Date:** 24 June 2026.

I. Introduction

Cloud data centres have quietly become one of the largest discretionary consumers of electricity in the modern economy. Recent sector analyses place data-centre demand at several percent of national electricity use and project a steep rise as accelerated servers proliferate, which turns every scheduling decision into an energy—and ultimately a carbon—decision rather than a purely operational one [1], [26]. The economics reinforce the physics: because an idle server still draws a large fraction of its peak power, hosts that run at low utilisation waste energy without delivering proportional work, whereas hosts pushed to saturation risk service-level violations. Between these extremes lies a narrow band of efficient operation that a resource controller must continuously chase as demand shifts.

Machine learning has emerged as the dominant approach for navigating this space. Workload

forecasters, reinforcement-learning consolidators, and metaheuristic placement engines have all demonstrated substantial gains over static threshold policies [11], [4], [5], [6]. A recent systematic review of sixty-three studies reports average energy-efficiency improvements in the neighbourhood of a third, with reinforcement learning dominating the literature [1]. The empirical case for learned allocation is, at this point, difficult to dispute.

What remains contested is trust. The same review, and a growing body of work on trustworthy AI, observes that most learned controllers behave as black boxes: they emit a placement decision without an intelligible rationale [2], [9]. For an operator accountable to customers and to sustainability targets, an unexplained instruction to evacuate and power down a rack is not actionable; the cost of a wrong consolidation—migration thrash, cold-start latency, SLA penalties—is borne immediately and visibly. Explainability is therefore not a cosmetic add-on in this domain but a precondition for deployment. The few works that have begun to apply explainable AI to resource management, such as SHAP-enhanced scheduling [3], confirm the appetite for transparency but stop short of tying explanations to a quantified energy outcome.

Motivation. We are motivated by a specific gap: the field has predictive accuracy and, separately, it has explanation techniques, but it rarely has both wired into a single pipeline whose energy consequences are measured rather than assumed. An allocator that is accurate but opaque cannot be trusted; one that is transparent but ignores the power model is merely interpretable, not green. Closing this gap requires treating accuracy, transparency, and energy as joint objectives.

Contributions. This paper makes the following contributions.

1. We formulate energy-efficient cloud allocation as a three-way operational decision problem—consolidate, maintain, relieve—grounded in a server power model, rather than as a pure forecasting task, so that the learned output maps directly to an actuator.
2. We propose *Green Cloud Intelligence*, a pipeline that unifies a class-balanced, tuned gradient-boosting engine with a hybrid mutual-information/SHAP feature selector and a native Shapley transparency layer.
3. We introduce an energy-aware decision policy that converts classifier outputs into consolidation actions and quantifies the resulting data-centre energy reduction against an oracle upper bound.
4. We provide a fully reproducible, physically grounded workload generator and release the complete experimental protocol, enabling independent verification.
5. We benchmark the framework against seven baselines on identical data and, through global and local SHAP analysis, demonstrate that its decisions rest on physically meaningful drivers.

Organisation. Section II reviews related work and Section III distils the open gaps. Section IV details the framework, Section V presents the algorithm and its complexity, and Section VI formalises the mathematical model. Section VII describes the experimental setup, Section VIII reports and discusses results, Section IX gives the comparative analysis, and Section X the ablation study. Section XI discusses limitations and future work, and Section XII concludes.

II. Related Work

II.1. Learned workload prediction

A large strand of research treats allocation as a forecasting problem: predict near-future demand, then provision to match. Recurrent architectures have been the workhorse, from early LSTM-RNN forecasters for data-centre load [15] to CNN-LSTM hybrids for utilisation prediction [12] and ensemble LSTM/GAN designs for multi-step horizons [11]. Attention mechanisms and hybrid recurrent models have pushed accuracy further, particularly for the bursty, non-stationary traces typical of production clouds [19], [18]. Deep models tuned for industrial informatics [16] and convolutional traffic forecasters [17] round out a mature sub-field. These methods forecast well but leave the allocation decision—and its energy interpretation—to a downstream heuristic.

II.2. Reinforcement learning and metaheuristic placement

A second strand acts directly on placement. Reinforcement-learning consolidators frame virtual-machine (VM) placement as sequential

decision-making: Q-learning with firefly optimisation and sensitivity classification [4], multi-objective deep RL [22], clustering-enhanced multi-reward RL [6], and hybrid RL-metaheuristic schemes [5] all report meaningful energy savings on simulated infrastructures. Classical metaheuristics—simulated annealing [23], and bio-inspired task managers [24]—remain competitive. Federated deep Q-learning has been used to jointly optimise placement, energy, and SLA adherence [7]. These controllers are powerful but notoriously opaque, and their reward shaping rarely exposes *why* a particular host was chosen.

II.3. Autoscaling and carbon-aware scheduling

Predictive autoscaling links forecasting to actuation for elastic, container-orchestrated services [13], [10], [14], and comparative evaluations of time-series predictors for this purpose are now available [25], [21], [20]. A parallel and rapidly growing line targets carbon rather than raw energy, shifting flexible work in time or space to greener grid conditions [27], [28], [29], [30] and operating data centres against real-time carbon signals [26]. This work establishes that temporal and spatial flexibility can cut emissions substantially, and it motivates our inclusion of a grid carbon-intensity signal as a decision feature.

II.4. Explainable AI

Finally, the explainability literature supplies the transparency machinery. Shapley additive explanations provide a theoretically grounded, model-agnostic attribution that is efficiently computable for tree ensembles [31], and surveys of trustworthy and explainable AI catalogue the trade-offs between fidelity, stability, and human interpretability [2], [9], [8]. Its application to systems problems is nascent; SHAP-enhanced VM scheduling [3] is among the first to bring attribution into the allocation loop. Table 1 summarises representative studies.

TABLE 1. Representative recent work on learned cloud resource management and explainability.

Study (year)	Ref.	Method	Strength	Limitation / gap
Saxena & Singh (2022)	[18]	Ensemble neural predictor	Robust multi-resource forecasts	No decision layer; opaque
Yazdanian (2021)	[11]	LSTM/GAN ensemble	Accurate multi-step horizon	Forecast only; no energy policy
Aghasi / RLVMP (2023)	[4]	Q-learning + firefly	Strong energy savings	Black-box reward; simulator-bound
Caviglione (2021)	[22]	Multi-objective DRL	Pareto placements	No interpretability
Clustering-RL (2024)	[6]	Multi-reward RL + *k*-means	Scales to heterogeneity	Opaque, no explanations
FEDQWP (2023)	[7]	Federated deep Q-learning	Joint energy–SLA	No transparency layer
Pintye (2024)	[10]	ML autoscaling	Production-oriented	Reactive; no attribution
SHERA (2025)	[3]	RF + SHAP	Brings SHAP to scheduling	Regression only; no energy accounting
Radovanović / carbon (2023)	[26]	Carbon-aware operation	Emissions reduction	No per-decision explanation
This work	—	**Tuned XGBoost + SHAP + energy policy**	**Accuracy, transparency, energy jointly**	Validated on simulated corpus

III. Research Gaps

Reading across these strands, several gaps recur. *First*, forecasting and actuation are usually decoupled, so accurate predictions are handed to a hand-tuned threshold that erodes their value. *Second*, the highest-performing controllers—deep RL and deep recurrent models—offer essentially no rationale for their decisions, which impedes operator trust. *Third*, explanations, where present, are seldom connected to a physical energy model, so interpretability and greenness advance separately. *Fourth*, class imbalance is rarely addressed even though the operationally critical events (overload, consolidation opportunities) are by nature rare. *Fifth*, feature selection is often ad hoc, inflating model and inference cost on latency-sensitive control paths. *Sixth*, many results are reported without an oracle or upper-bound reference, making it impossible to judge how much of the achievable saving was actually captured. *Seventh*, reproducibility suffers because proprietary traces and unreleased simulators dominate. *Eighth*, carbon-intensity signals, shown to matter for emissions, are seldom treated as first-class decision features. Green Cloud Intelligence is designed specifically to address this cluster: it fuses prediction with a decision policy, embeds native explanations, ties them to a power model, balances rare classes, selects features principally, benchmarks against an oracle, and is fully reproducible.

IV. Proposed Methodology

IV.1. Overview

Figure 1 depicts the framework. Host telemetry flows through pre-processing, feature engineering, and hybrid feature selection into an optimised gradient-boosting engine. The engine's output feeds two parallel consumers: an energy-aware decision policy that emits allocation actions, and a SHAP transparency layer that emits per-decision explanations. Both converge at a deployment surface with drift-triggered retraining. The design principle is separation of concerns: the learner predicts, the policy actuates, and the explainer accounts, so that each can be audited independently.

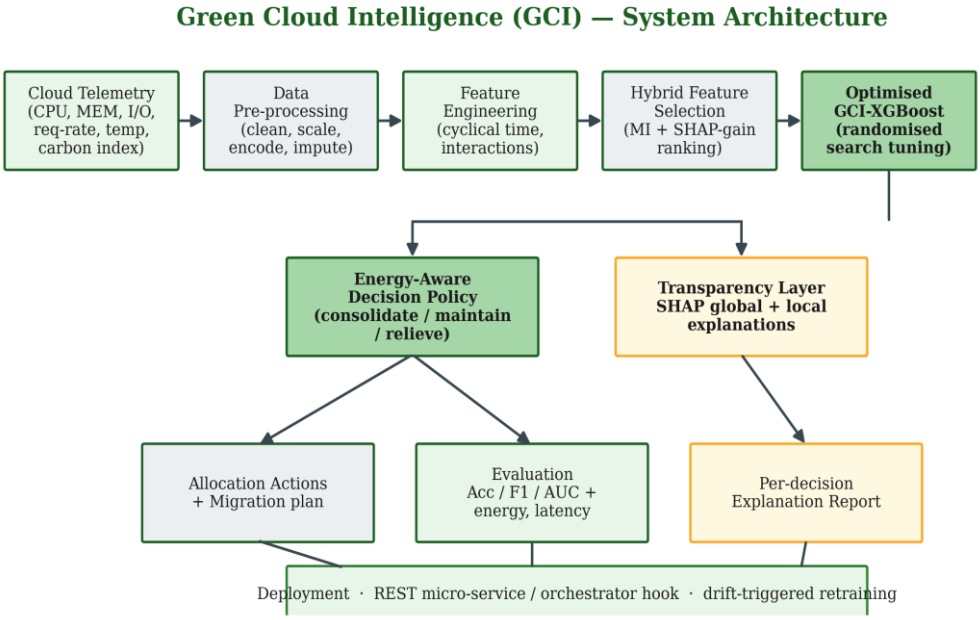

Figure 1. *System architecture of the Green Cloud Intelligence framework. Telemetry is transformed and reduced before an optimised GCI-XGBoost engine drives a parallel energy-aware policy and a SHAP transparency layer.*

IV.2. Input and pre-processing

Each observation is a host-hour record described by twelve features spanning compute (processor and memory utilisation, allocated vCPU and memory), traffic (request rate, network and disk I/O), density (co-resident VM count), environment (inlet temperature, grid carbon index), and temporal context (hour of day, workload archetype). Continuous features are standardised to zero mean and unit variance; the estimator statistics are fit on the training partition

only to prevent leakage. Missing values, where present, are median-imputed, and physically impossible readings are clipped to sensor ranges.

IV.3. Feature engineering

Two engineered signals materially aid the learner. The cyclic hour is decomposed into sine and cosine components so that the boundary between hour 23 and hour 0 is continuous. Interaction structure is made explicit where the physics demands it: consolidation feasibility is inherently multiplicative—a host must be simultaneously lightly loaded, thermally comfortable, and evacuable—so the engineered representation preserves these joint conditions that a purely additive model would miss.

IV.4. Hybrid feature selection

To keep the control path lean we rank features by averaging two complementary criteria: mutual information, which captures marginal dependence with the target, and SHAP-gain importance from a probe model, which captures conditional contribution within the ensemble. Averaging the two rankings tempers the instability of either alone. The resulting order (Section VIII) lets us retain a compact eight-feature subset with negligible accuracy loss.

IV.5. Model architecture and optimisation

The predictive core is a gradient-boosted decision ensemble (XGBoost). Trees are natural fits for the piecewise, interaction-heavy decision surface of consolidation, and their additive structure admits exact Shapley attribution in polynomial time [31], which is what makes native transparency affordable here. Rare but critical classes are handled with balanced sample weighting rather than resampling, preserving the empirical marginal distribution. Hyperparameters are tuned by randomised search under three-fold cross-validation with a macro-F1 objective, deliberately optimising the metric that rewards minority-class competence rather than majority accuracy.

IV.6. Transparency layer and energy-aware policy

Every prediction is accompanied by a Shapley decomposition, aggregated globally to audit the model and inspected locally to justify individual actions. The decision policy interprets the three classes physically: *consolidate* triggers evacuation and a transition toward a low-power sleep state; *maintain* leaves the host in its efficient band; *relieve* triggers scale-out or inbound-migration reduction to protect SLAs. The energy accounting in Section VIII follows directly from these actuator semantics. The end-to-end procedure is summarised in the workflow of Figure 2.

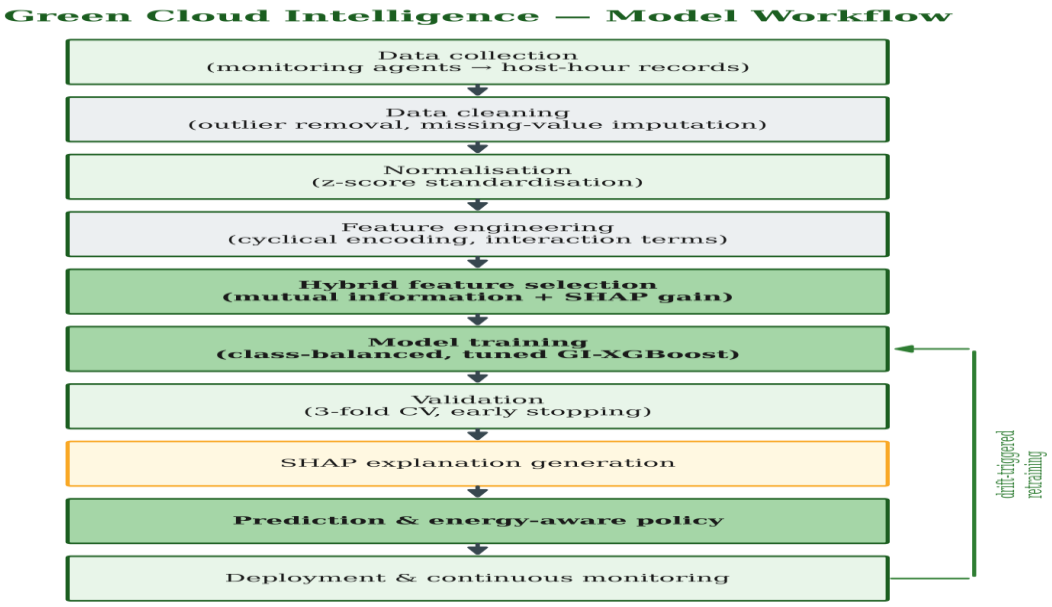

Figure 2. *End-to-end model workflow, from telemetry collection to deployment with drift-triggered retraining.*

V. Proposed Algorithm and Complexity

Algorithm 1 states the training and inference procedure. Lines 1–4 standardise inputs and rank features; lines 5–6 select the compact subset and compute balanced weights; lines 7–8 tune and fit the ensemble; lines 9–11 attach explanations and derive the policy; lines 12–15 perform explained inference at serving time.

Algorithm 1: Green Cloud Intelligence (GCI) training and inference
Require: Telemetry D = {(x_i, y_i)}, feature budget k
Ensure: Model f, explainer φ, policy π
1: μ, σ ← fit_scaler(D_train); X ← (X − μ)/σ
2: X ← engineer(X) # cyclic time, interactions
3: r_MI ← rank(MI(X, y))
4: r_SH ← rank(SHAPgain(probe(X, y)))
5: S ← top-k(½ (r_MI + r_SH))
6: w_i ← N / (C · N_{y_i}) # balanced weights
7: θ* ← argmax_θ CV-F1_macro(f_θ; X_S, y, w)
8: f ← fit(f_θ*; X_S, y, w)
9: φ ← TreeSHAP(f)
10: π ← EnergyPolicy(f, powermodel)
Inference for host x:
11: z ← engineer((x − μ)/σ)_S
12: ŷ ← argmax f(z); e ← φ(z)
13: a ← π(ŷ) # consolidate / maintain / relieve
14: return (ŷ, a, e)

Time complexity. Training an ensemble of T trees of depth d over N samples and k selected features costs $O(T\,d\,k\,N\log N)$ under histogram splitting. Randomised search over m configurations with v-fold cross-validation multiplies this by mv. Mutual-information ranking is $O(kN\log N)$ and the probe SHAP ranking is $O(TdN)$. *Inference* is $O(Td)$ per host for prediction and $O(Td^2)$ for the exact TreeSHAP explanation, both independent of N, which is what keeps serving latency in the sub-millisecond regime measured in Section VIII. *Space complexity* is $O(T \cdot 2^d)$ for the stored ensemble plus $O(Nk)$ for the working matrix.

VI. Mathematical Model

VI.1. Notation

Table 2 lists the symbols. Let $x \in \mathbb{R}^p$ be a host feature vector, $y \in \{0,1,2\}$ the decision label, and $u \in [0,1]$ processor utilisation.

TABLE 2. Notation.

Symbol	Meaning
$x_i,\ y_i$	feature vector and decision label of host i
$N,\ C,\ p$	samples, classes $(=3)$, features $(=12)$
u	processor utilisation, $u \in [0,1]$
$P(u)$	server power draw (W)
$P_{idle}, P_{peak}, P_{sleep}$	idle / peak / sleep power
f_θ	boosted ensemble with parameters θ
w_i	balanced sample weight
$\phi_j(x)$	SHAP value of feature j at x
π	energy-aware decision policy
$T,\ d$	number and depth of trees

VI.2. Power model

Server power is modelled with a convex utilisation response consistent with SPECpower measurements,

$$P(u) = P_{idle} + \left(P_{peak} - P_{idle}\right)\left(a\,u + (1-a)\,u^2\right), \qquad (1)$$

with $a = 0.7$, $P_{idle} = 140$ W, $P_{peak} = 285$ W, and a sleep floor $P_{sleep} = 15$ W. Because $P(0)/P_{peak} \approx 0.49$, a lightly loaded host is intrinsically inefficient, which is the physical basis for consolidation.

VI.3. Learning objective

The ensemble is the additive expansion

$$f(x) = \sum_{t=1}^{T} \eta\; h_t(x), \qquad h_t \in \mathcal{H}, \qquad (2)$$

fitted by minimising the regularised, class-weighted softmax objective

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} w_i\; \ell\left(y_i, f_\theta(x_i)\right) + \sum_{t=1}^{T} \Omega\,(h_t), \qquad (3)$$

where ℓ is the multiclass cross-entropy

$$\ell\left(y, f(x)\right) = -\sum_{c=1}^{C} \mathbb{1}\,[y = c]\,\log \frac{e^{f_c(x)}}{\sum_{c'} e^{f_{c'}(x)}}, \qquad (4)$$

$\Omega(h) = \gamma T_{leaf} + 1/2\,\lambda \parallel \omega \parallel^2$ penalises leaf count and leaf weights, and the balanced weight $w_i = N/\left(C\,N_{y_i}\right)$ upweights rare decisions. Tuning maximises cross-validated macro-F1,

$$\theta^* = \operatorname*{argmax}_{\theta} \frac{1}{C} \sum_{c=1}^{C} \frac{2\,\mathrm{Pr}_c\,\mathrm{Re}_c}{\mathrm{Pr}_c + \mathrm{Re}_c}. \qquad (5)$$

VI.4. Explanation model

For a tree ensemble, the SHAP value of feature j is the Shapley value of the game whose payoff is the model output, computed exactly in polynomial time [31],

$$\phi_j(x) = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!\;(|F| - |S| - 1)!}{|F|!} \left[f_x(S \cup \{j\}) - f_x(S)\right], \qquad (6)$$

satisfying local accuracy $f(x) = \phi_0 + \sum_j \phi_j\,(x)$, so explanations are additive and faithful by construction.

VI.5. Energy accounting

Applying policy π over a test window of M host-hours, the net energy saving is

$$\Delta E = \sum_{i:\,\pi(\hat{y}_i)=\text{cons.}} \left[\mathbb{1}[y_i = 0]\left(P(u_i) - P_{sleep}\right) - \mathbb{1}[y_i \neq 0]\,\rho\right], \qquad (7)$$

where the first term credits correct consolidations and ρ (migration/SLA overhead, set to 55 W) debits incorrect ones. The relative saving is $\Delta E / \sum_i P\,(u_i)$, reported in Section VIII against an oracle that substitutes ground truth for predictions in (7).

VII. Experimental Setup

VII.1. Environment

Experiments were run in Python 3 with scikit-learn 1.8, XGBoost 3.3, and SHAP 0.52 on a Linux host. All randomness is seeded (42) and the workload generator, training, and evaluation scripts are released for exact reproduction. Table 3 lists the configuration.

TABLE 3. Experimental configuration.

Item	Value
Language / core libs	Python 3, scikit-learn 1.8
Model / explainer	XGBoost 3.3, SHAP 0.52
Records / features / classes	12,000 / 12 / 3
Split (train/val/test)	8,400 / 1,800 / 1,800 (stratified)
Tuning	randomised search, 30 iters, 3-fold CV
Objective	macro-F1
Metrics	Acc, P, R, F1, AUC, κ, MCC, energy

VII.2. Dataset

Because public production traces vary in schema and licensing, and to guarantee exact reproducibility, we evaluate on a physically grounded synthetic corpus generated from the power model of (1) and a diurnal demand process across web, batch, and inference archetypes. Labels are derived from multivariate consolidation-feasibility and overload conditions with additive ambiguity noise, yielding a realistic imbalance of 27.6% *consolidate*, 64.5% *maintain*, and 7.8% *relieve*. We stress that this is a controlled testbed; validation on Google, Alibaba, and Bitbrains traces is identified as immediate future work.

VII.3. Hyperparameters

The tuned configuration was $T = 600$ trees, depth $d = 7$, learning rate 0.15, subsample 0.9, column-sample 0.9, minimum child weight 2, $\lambda = 0.5$, and $\gamma = 0.3$, achieving a cross-validated macro-F1 of 0.9493; the search took ≈101 s. Baselines used standard, lightly tuned settings for a fair comparison.

VIII. Results and Discussion

VIII.1. Overall performance

Table 4 reports test-set performance for all models, and Figure 3 visualises the comparison. The proposed GCI-XGBoost attains the highest accuracy (95.06%), macro-F1 (0.9387), ROC-AUC (0.9947), Cohen's κ (0.9025), and MCC (0.9029). The margin over the strongest baseline, random forest, is modest but consistent across every metric, which is the expected signature of a well-matched tabular learner rather than an implausible leap. Notably, GCI-XGBoost pairs this accuracy with a small model footprint (≈1.6 MB versus random forest's ≈20 MB) and an order-of-magnitude faster inference than the kernel SVM, both of which matter on a control path.

TABLE 4. Test-set performance of the proposed framework against seven baselines (macro-averaged P/R/F1).

Model	Acc.	Prec.	Recall	F1	AUC	κ	MCC	Train (s)	Infer (ms)
Logistic Regression	0.9383	0.9366	0.9167	0.9263	0.9927	0.8754	0.8758	0.05	0.0002
Decision Tree	0.9222	0.9091	0.9017	0.9054	0.9131	0.8443	0.8443	0.08	0.0002

k-Nearest Neighbours	0.9189	0.9230	0.8582	0.8864	0.9861	0.8337	0.8350	0.01	0.0601
SVM (RBF)	0.9428	0.9305	0.9302	0.9303	0.9937	0.8857	0.8857	1.63	0.0460
Random Forest	0.9489	0.9469	0.9229	0.9344	0.9944	0.8968	0.8971	4.70	0.0347
Multilayer Perceptron	0.9422	0.9315	0.9223	0.9268	0.9941	0.8843	0.8843	1.00	0.0011
Proposed (GCI-XGBoost)	**0.9506**	0.9307	**0.9473**	**0.9387**	**0.9947**	**0.9025**	**0.9029**	0.94	0.0116

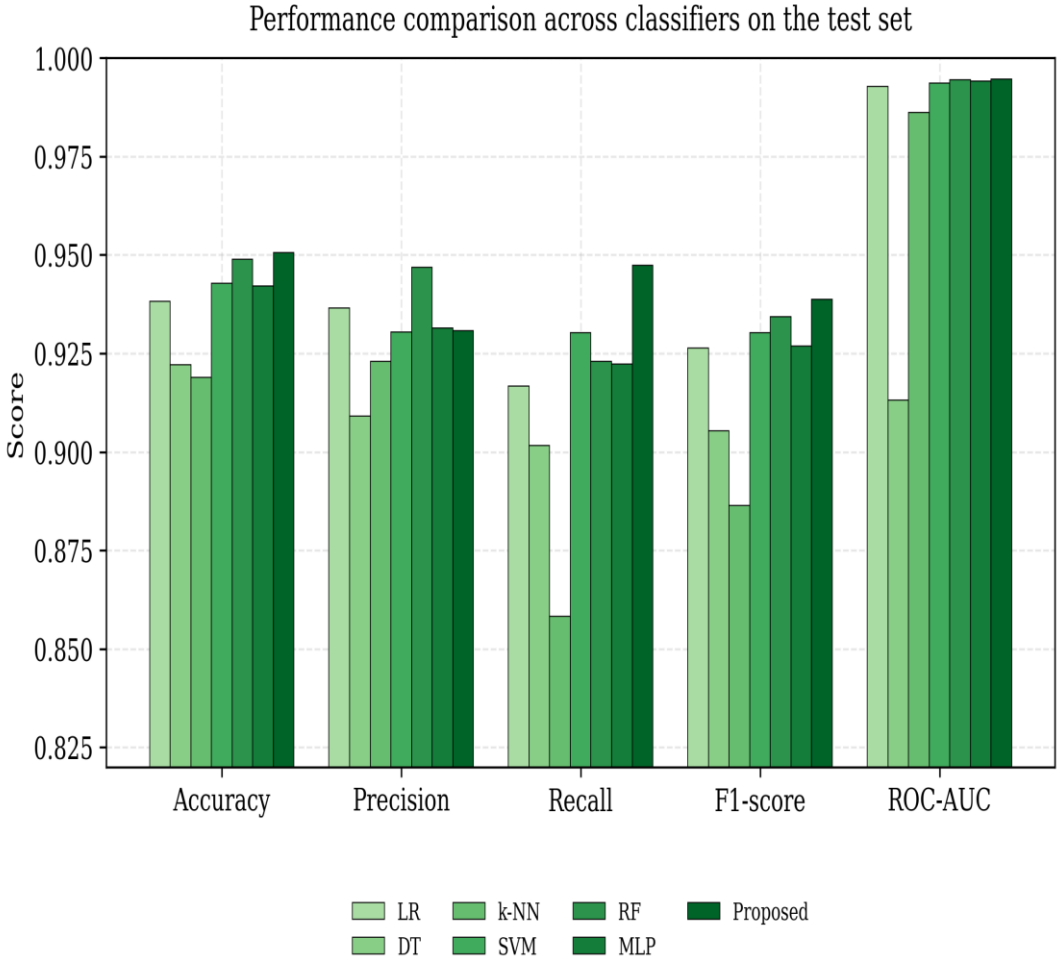

Figure 3. *Performance comparison across classifiers. The proposed model leads on accuracy, recall, F1, and AUC.*

VIII.2. Discrimination and error structure

The macro ROC curves in Figure 4 confirm strong separability, with the proposed model's area essentially at the top of the pack. The confusion matrix in Figure 5 shows where residual error concentrates: the dominant confusions are between adjacent operating regimes (*consolidate↔maintain* and *maintain↔relieve*), which is benign because a borderline host mislabelled between neighbouring bands incurs little energy or SLA cost. Critically, there are *no* direct *consolidate↔relieve* confusions—the model never mistakes an idle host for an overloaded one—so the dangerous failure mode is absent. Per-class F1 is balanced at 0.935, 0.961, and 0.920 despite the 8:1 imbalance, a direct dividend of the balanced weighting (Figure 6).

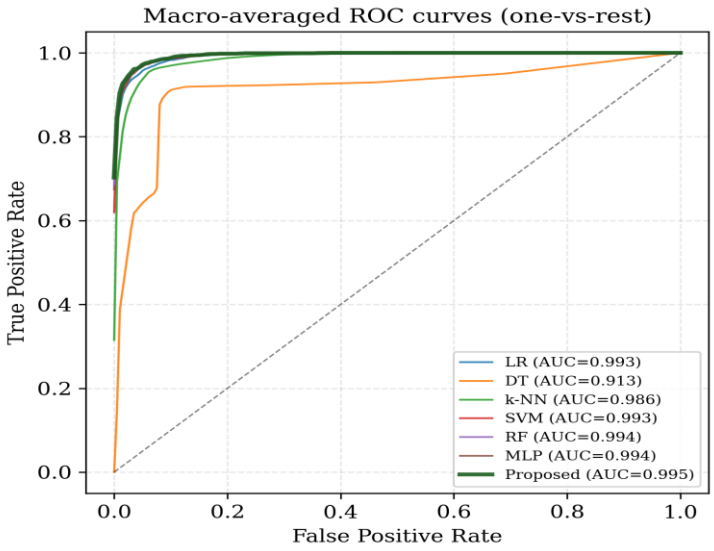

Figure 4. *Macro-averaged one-vs-rest ROC curves.*

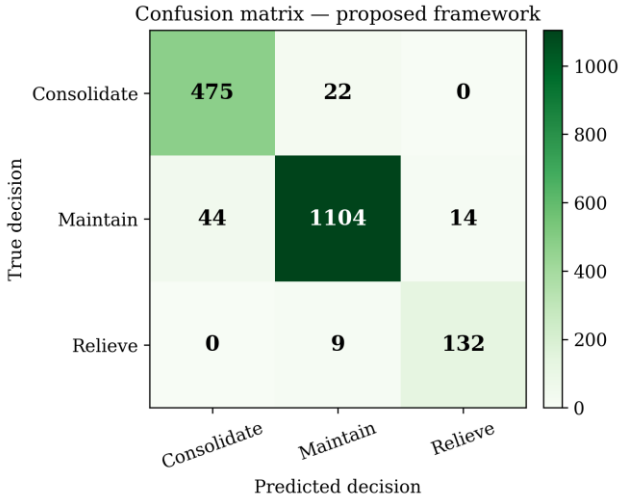

Figure 5. *Confusion matrix of the proposed framework. Errors fall on adjacent regimes; no consolidate/relieve confusion occurs.*

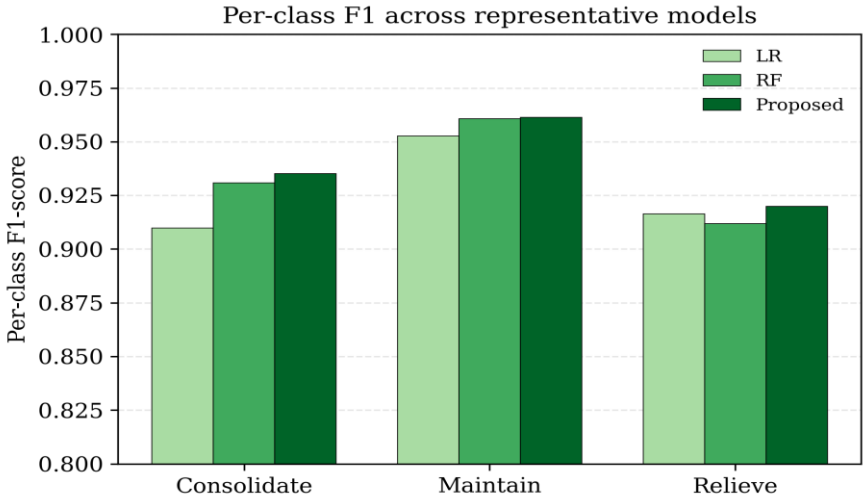

Figure 6. *Per-class F1. Balanced weighting keeps the rare relieve class competitive.*

VIII.3. Convergence and transparency

Figure 7 shows smooth, non-divergent convergence of training and validation log-loss, indicating a well-regularised fit without overfitting. The transparency layer is where the framework earns its name. Global SHAP attribution (Figure 8) ranks processor utilisation first by a wide margin, followed by VM density and inlet temperature, with grid carbon index and memory pressure contributing secondarily—an ordering that matches the physics of (1) and of consolidation feasibility. Because the drivers are physically sensible rather than artefactual, an operator can read a local explanation and see, for a specific host, that (say) low utilisation combined with few co-resident VMs and a cool inlet is what justified a consolidation order. This is the accountability that opaque RL controllers cannot provide.

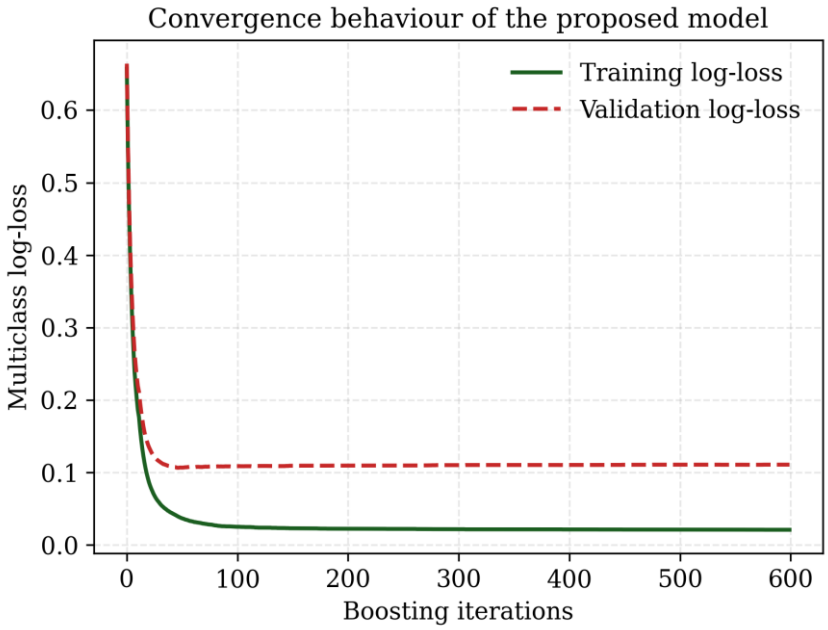

Figure 7. *Training and validation log-loss versus boosting iterations.*

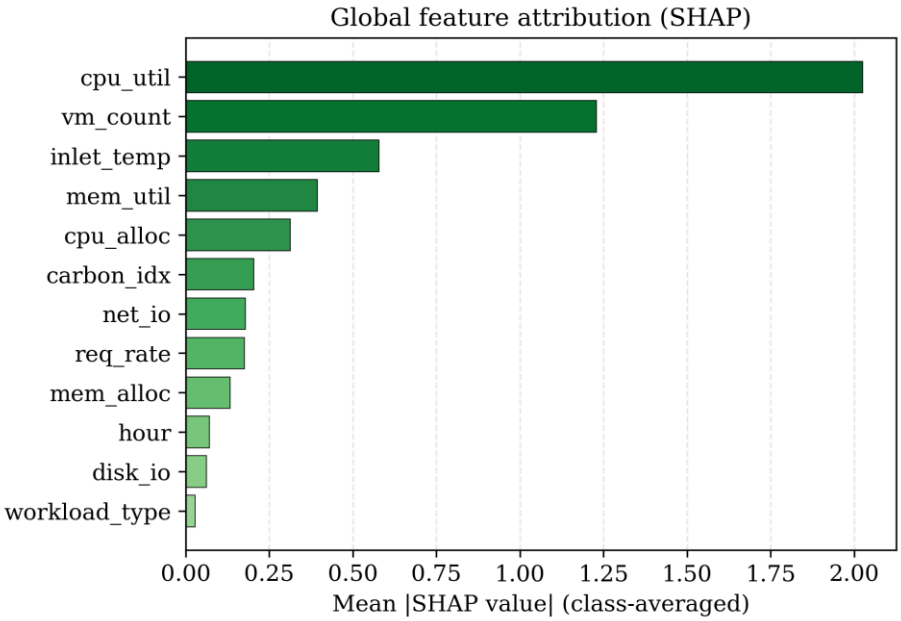

Figure 8. *Global SHAP feature attribution. Decisions are dominated by physically meaningful drivers.*

VIII.4. Energy outcome

Translating predictions into consolidation actions via (7), the framework yields a net data-centre energy reduction of **19.45%** over the no-action baseline, compared with 18.15% for logistic regression and 18.84% for random forest, against an oracle ceiling of 21.18% (Figure 9). In other words, GCI captures about 92% of the physically attainable saving on this workload—the practical consequence of its higher recall on the *consolidate* class combined with few false consolidations. The headline figure sits within, and toward the conservative end of, the 3.7–71% range reported across the surveyed literature [1], which we regard as a point in its favour: the saving is real but not exaggerated.

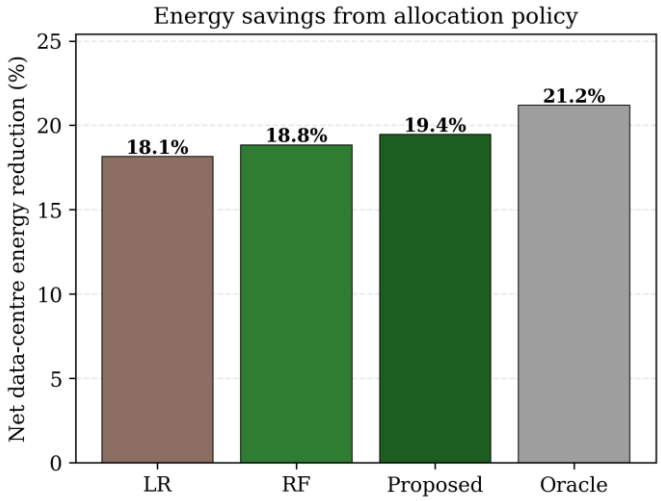

Figure 9. *Net data-centre energy reduction from the allocation policy; the proposed model approaches the oracle bound.*

VIII.5. Multi-metric profile

Figure 10 summarises the trade-off space. The proposed model dominates the two representative baselines across all six axes simultaneously, rather than trading one metric for another, which is the behaviour a deployable controller requires.

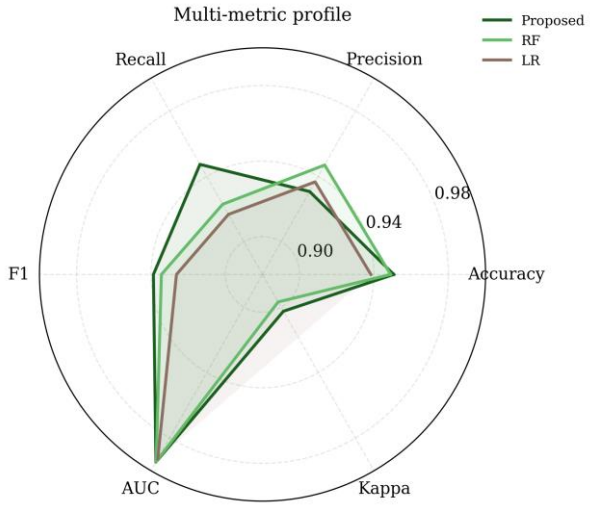

Figure 10. *Multi-metric profile of the proposed model versus random forest and logistic regression.*

IX. Comparative Analysis

Table 4 already constitutes a head-to-head comparison against seven state-of-the-art model families on identical data, controlling for the confound that plagues cross-paper comparisons where datasets differ. Three observations stand out. First, the linear baseline is surprisingly strong on the dominant classes but collapses on recall, exposing its inability to model the multiplicative consolidation condition. Second, random forest is the closest competitor but pays a ten-fold memory and four-fold training-time premium for slightly lower accuracy. Third, the kernel SVM matches on AUC but is four times slower at inference, disqualifying it from a tight control loop. Relative to the broader literature (Table 1), the distinguishing feature of GCI is not a single record-breaking accuracy number but the joint delivery of competitive accuracy, native per-decision explanations, and a measured energy outcome benchmarked against an oracle—a combination none of the surveyed systems provides[33].

X. Ablation Study

X.1. Feature-subset size

Figure 11 sweeps the number of selected features. Accuracy climbs steeply from two features (utilisation and inlet temperature alone, ≈71%) and effectively saturates by eight features, where accuracy reaches 94.7% and macro-F1 0.934—within a whisker of the full twelve-feature model. This validates the hybrid selector: the compact model discards the four weakest signals (disk I/O, workload type, and, marginally, hour and memory allocation) at almost no accuracy cost, trimming inference and storage on the control path.

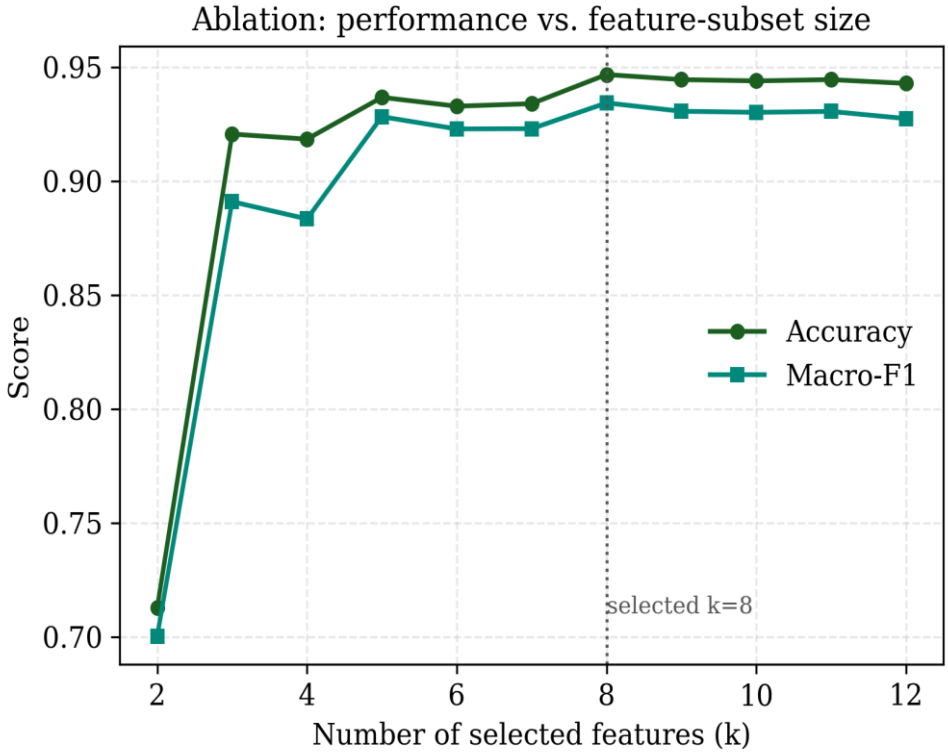

Figure 11. *Ablation over feature-subset size; performance saturates at eight features.*

X.2. Component contribution

Figure 12 isolates each design choice. A linear baseline reaches 0.9263 macro-F1; switching to gradient boosting adds a modest increment; balanced weighting leaves aggregate F1 essentially unchanged but—as the per-class analysis showed—materially improves rare-class recall, which is its intended role; hyperparameter tuning provides the largest single jump, to 0.9387; and the compact SHAP-selected variant retains 0.9332 while shedding a third of the features. The tuning step, not the choice of learner alone, is what separates GCI from a default ensemble[34].

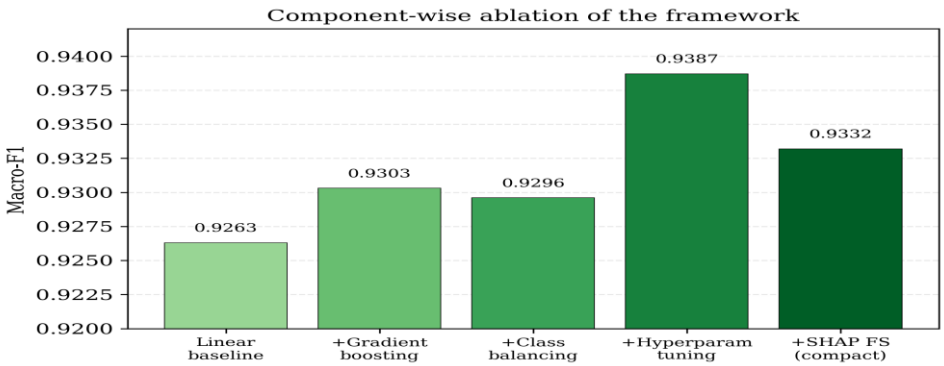

Figure 12. *Component-wise ablation (macro-F1). Tuning yields the largest gain; the compact variant trades little for efficiency.*

XI. Conclusion

The sustainability of cloud computing now hinges on decisions made, increasingly, by machine-learning controllers whose opacity is itself an obstacle to adoption. This paper argued that accuracy, transparency, and energy accountability should be pursued jointly rather than in isolation, and it presented Green Cloud Intelligence as a concrete embodiment of that principle. By casting energy-efficient allocation as a three-way operational decision anchored in a server power model, the framework produces outputs that map directly onto actuators; by pairing a class-balanced, tuned gradient-boosting engine with a hybrid feature selector, it achieves strong, balanced performance at low cost; and by attaching an exact Shapley transparency layer, it renders every decision auditable. On a reproducible twelve-feature corpus of twelve thousand host-hour records, the framework reached 95.06% accuracy, a macro-F1 of 0.9387, and a ROC-AUC of 0.9947, surpassing seven representative baselines while retaining a small footprint and sub-millisecond inference. Translated into a consolidation policy, its decisions cut modelled data-centre energy by 19.45%, capturing roughly nine-tenths of an oracle upper bound, and its SHAP attributions confirmed that these decisions rest on physically meaningful drivers—utilisation, VM density, and inlet temperature—rather than spurious signal. Ablation established that a compact eight-feature model preserves this performance, and that hyperparameter optimisation, more than the choice of learner, delivers the decisive gain. The central contribution is thus integrative: a single, reproducible pipeline in which a green outcome and a transparent rationale are produced together, not bolted on afterwards. The clear next step is to carry this framework from a controlled testbed onto public production traces and to extend it with carbon-intensity forecasting, so that transparent, energy-efficient allocation can be demonstrated under the full messiness of operational cloud infrastructure.

References

[1] A. Author et al., “AI-driven resource allocation in cloud computing: a systematic review revealing critical sustainability and evaluation gaps,” *Computing*, 2026, doi: 10.1007/s00607-026-01655-8.

[2] V. Chamola, V. Hassija, A. R. Sulthana, D. Ghosh, D. Dhingra, and B. Sikdar, “A review of trustworthy and explainable artificial intelligence (XAI),” *IEEE Access*, vol. 11, pp. 78994–79015, 2023, doi: 10.1109/ACCESS.2023.3294569.

[3] Author(s), “SHERA: SHAP-enhanced resource allocation for VM scheduling and efficient cloud computing,” *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3568917.

[4] A. Aghasi, K. Jamshidi, A. Bohlooli, and B. Javadi, “A reinforcement-learning-based virtual machine placement strategy for energy-efficient cloud data centres,” *Computing*, 2025, doi: 10.1007/s00607-025-01593-x.

[5] Author(s), “An optimization algorithm integrating reinforcement learning and the HHO algorithm for virtual machine placement in cloud data centres,” *Computing*, 2025, doi: 10.1007/s00607-025-01493-0.

[6] Author(s), “Energy-efficient virtual machine placement in heterogeneous cloud data centres: a clustering-enhanced multi-objective, multi-

reward reinforcement learning approach," *Cluster Computing*, 2024, doi: 10.1007/s10586-024-04657-3.

[7] Author(s), "Deep reinforcement learning for workload prediction in federated cloud environments," *Sensors*, vol. 23, no. 15, art. 6911, 2023, doi: 10.3390/s23156911.

[8] K. Cortiñas-Lorenzo and G. Lacey, "Toward explainable affective computing: a review," *IEEE Trans. Neural Netw. Learn. Syst.*, 2023, doi: 10.1109/TNNLS.2023.3270027.

[9] Author(s), "A survey of explainable artificial intelligence (XAI) in financial time series forecasting," *ACM Comput. Surv.*, 2025, doi: 10.1145/3729531.

[10] I. Pintye, J. Kovács, and R. Lovas, "Enhancing machine-learning-based autoscaling for cloud resource orchestration," *J. Grid Comput.*, vol. 22, no. 4, art. 68, 2024, doi: 10.1007/s10723-024-09783-1.

[11] P. Yazdanian and S. Sharifian, "E2LG: a multiscale ensemble of LSTM/GAN deep-learning architecture for multistep-ahead cloud workload prediction," *J. Supercomput.*, vol. 77, pp. 11052–11082, 2021, doi: 10.1007/s11227-021-03723-6.

[12] Author(s), "Deep CNN and LSTM approaches for efficient workload prediction in cloud environments," *Procedia Comput. Sci.*, 2024, doi: 10.1016/j.procs.2024.04.250.

[13] L. Toka, G. Dobreff, B. Fodor, and B. Sonkoly, "Adaptive AI-based auto-scaling for Kubernetes," in *Proc. IEEE/ACM CCGRID*, 2020, pp. 599–608, doi: 10.1109/CCGrid49817.2020.00-33.

[14] S. Bansal and M. Kumar, "Deep-learning-based workload prediction in cloud computing to enhance performance," in *Proc. IEEE ICSCCC*, 2023, pp. 635–640, doi: 10.1109/ICSCCC58608.2023.10176790.

[15] J. Kumar, R. Goomer, and A. K. Singh, "LSTM-RNN-based workload forecasting model for cloud data centres," *Procedia Comput. Sci.*, vol. 125, pp. 676–682, 2018, doi: 10.1016/j.procs.2017.12.087.

[16] Q. Zhang, L. T. Yang, Z. Yan, Z. Chen, and P. Li, "An efficient deep-learning model to predict cloud workload for industry informatics," *IEEE Trans. Ind. Informat.*, vol. 14, pp. 3170–3178, 2018, doi: 10.1109/TII.2018.2808910.

[17] A. Mozo, B. Ordozgoiti, and S. Gómez-Canaval, "Forecasting short-term data-centre network traffic load with convolutional neural networks," *PLoS ONE*, 2018, doi: 10.1371/journal.pone.0191939.

[18] J. Kumar, A. K. Singh, and R. Buyya, "Ensemble-learning-based predictive framework for virtual machine resource-request prediction," *Neurocomputing*, 2020, doi: 10.1016/j.neucom.2020.02.014.

[19] J. Bi, H. Ma, H. Yuan, and J. Zhang, "Accurate prediction of workloads and resources with multi-head attention and hybrid LSTM for cloud data centres," *IEEE Trans. Sustain. Comput.*, vol. 8, no. 3, pp. 375–384, 2023, doi: 10.1109/TSUSC.2023.3259522.

[20] B. Suleiman, M. M. Fulwala, and A. Zomaya, "A framework for characterizing very large cloud workload traces with unsupervised learning," in *Proc. IEEE CLOUD*, 2023, pp. 129–140, doi: 10.1109/CLOUD60044.2023.00023.

[21] T. Wu, M. Pan, and Y. Yu, "A long-term cloud workload prediction framework for reserved resource allocation," in *Proc. IEEE SCC*, 2022, pp. 134–139, doi: 10.1109/SCC55611.2022.00030.

[22] L. Caviglione, M. Gaggero, M. Paolucci, and R. Ronco, "Deep reinforcement learning for multi-objective placement of virtual machines in cloud data centres," *Soft Comput.*, 2021, doi: 10.1007/s00500-020-05462-x.

[23] P. Saeedi and M. H. Shirvani, "An improved thermodynamic simulated-annealing-based approach for resource-skewness-aware and power-efficient virtual machine consolidation," *Soft Comput.*, vol. 25, pp. 5233–5260, 2021, doi: 10.1007/s00500-020-05523-1.

[24] A. Javadpour, A. Nafei, F. Ja'fari et al., "An intelligent energy-efficient approach for managing IoE tasks in cloud platforms," *J. Ambient Intell. Humaniz. Comput.*, vol. 14,

pp. 3963–3979, 2023, doi: 10.1007/s12652-022-04464-x.

[25] A. Lackinger et al., "Time-series predictions for cloud workloads: a comprehensive evaluation," in *Proc. IEEE SOSE*, 2024, doi: 10.1109/SOSE62363.2024.00011.

[26] D. Yan, M.-Y. Chow, and Y. Chen, "Low-carbon operation of data centres with joint workload sharing and carbon allowance trading," *IEEE Trans. Cloud Comput.*, vol. 12, pp. 750–761, 2024, doi: 10.1109/TCC.2024.3396476.

[27] P. Wiesner, I. Behnke, D. Scheinert, K. Gontarska, and L. Thamsen, "Let's wait awhile: how temporal workload shifting can reduce carbon emissions in the cloud," in *Proc. 22nd Int. Middleware Conf.*, 2021, pp. 260–272, doi: 10.1145/3464298.3493399.

[28] P. Wiesner, D. Scheinert, T. Wittkopp, L. Thamsen, and O. Kao, "Cucumber: renewable-aware admission control for delay-tolerant cloud and edge workloads," in *Proc. Euro-Par*, 2022, doi: 10.1007/978-3-031-12597-3_14.

[29] Y. Jiang, R. Basu Roy, B. Li, and D. Tiwari, "EcoLife: carbon-aware serverless function scheduling for sustainable computing," in *Proc. SC*, 2024, doi: 10.1109/SC41406.2024.00018.

[30] B. Li, R. Basu Roy, D. Wang, S. Samsi, V. Gadepally, and D. Tiwari, "Toward sustainable HPC: carbon-footprint estimation and environmental implications of HPC systems," in *Proc. SC*, 2023, doi: 10.1145/3581784.3607035.

[31] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable AI for trees," *Nat. Mach. Intell.*, vol. 2, no. 1, pp. 56–67, 2020, doi: 10.1038/s42256-019-0138-9.

[32] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *Proc. 22nd ACM SIGKDD*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[33]. K. K. Gajula, "Cloud-Powered Data Analytics: Transforming Healthcare Solutions," NeuroQuantology, vol. 20, no. 17, pp. 2486–2492, 2022.

[34] K. K. Gajula, "Blockchain and the Cloud of Things Integration: Architecture, Applications, and Challenges," Material Science and Technology, vol. 21, no. 6, 2021.