IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, June 2026
Impact Factor: 8.2

Machine Learning-Based Intelligent Resource Allocation Framework for Dynamic Cloud Computing Environments

Gaurav Bhakuni[1] and Dr. Sandeep Kumar[2]
M. Tech Scholar, Department of Computer Science and Engineering
Associate Professor, Department of Computer Science and Engineering
Tula's Institute, Dehradun, Uttarakhand, India
gauravuk0808@gmail.com and sandeepkumarsiet@gmail.com

Abstract: *Cloud computing has become the backbone of modern digital services, yet the highly dynamic and unpredictable nature of user workloads continues to make efficient resource allocation a persistent challenge. Conventional rule-based and heuristic provisioning strategies, such as round-robin, static thresholds, and metaheuristic optimizers, react to load changes only after they occur and therefore lead to either over-provisioning, which wastes energy and capital, or under-provisioning, which violates service-level agreements (SLAs). This paper proposes a Machine Learning-based Resource Allocation Framework (ML-RAF) that couples a Long Short-Term Memory (LSTM) workload forecaster with a Deep Reinforcement Learning (DRL) allocation agent to provision computing resources proactively in dynamic cloud environments. The LSTM module predicts short-horizon demand for CPU, memory, and bandwidth from historical monitoring traces, while the DRL agent learns an allocation policy that jointly optimizes resource utilization, response time, energy consumption, and SLA compliance. The framework was implemented and evaluated on a CloudSim-based simulation driven by a realistic workload trace, and benchmarked against four baselines: Round Robin, Threshold-based scaling, a Genetic Algorithm, and Particle Swarm Optimization. Experimental results show that the proposed ML-RAF improves average CPU utilization to 93%, reduces average response time by up to 67%, lowers data-center energy consumption by up to 37%, and decreases the SLA violation rate to 2.3%, clearly outperforming all baseline techniques. The findings confirm that combining predictive forecasting with learning-based decision making offers a practical and scalable path toward autonomous, energy-aware cloud resource management.*

Keywords: Cloud Computing; Resource Allocation; Machine Learning; Deep Reinforcement Learning; LSTM; Workload Prediction; Auto-Scaling; Energy Efficiency; Service-Level Agreement; CloudSim

I. INTRODUCTION

Cloud computing has fundamentally reshaped the way computational resources are consumed and delivered. By offering on-demand access to a shared pool of configurable computing resources, including servers, storage, networks, and applications, cloud platforms allow organizations to scale services elastically while paying only for what they use. This elasticity, however, comes with a significant operational burden: cloud providers must continuously decide how much capacity to allocate to each tenant and workload, in real time, under conditions of considerable uncertainty.

The workloads handled by contemporary data centers are highly heterogeneous and non-stationary. Web traffic exhibits diurnal and weekly seasonality punctuated by sudden flash crowds; batch analytics jobs arrive in irregular bursts; and latency-sensitive microservices coexist with throughput-oriented background tasks on the same physical infrastructure. In such an environment, the central question of resource allocation, namely how many virtual machines (VMs) or

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, June 2026
Impact Factor: 8.2

containers to provision, where to place them, and when to scale them up or down, becomes both critical and extraordinarily difficult.

Traditional resource allocation mechanisms fall broadly into three categories. Static allocation reserves a fixed amount of capacity per application and is simple but wasteful, since it must be sized for peak demand. Rule-based or threshold-based auto-scaling reacts to monitored metrics, for example by adding a VM when average CPU utilization exceeds 80%, but it is inherently reactive: by the time the threshold is breached, the SLA may already be at risk, and the lag introduced by VM boot time aggravates the problem. Metaheuristic optimizers such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) can find high-quality allocations for a given snapshot of demand, but they treat each allocation as an isolated optimization problem and do not learn from the temporal structure of the workload.

The shortcomings of these approaches share a common root cause: they are reactive rather than predictive, and they do not improve with experience. A more attractive paradigm is to anticipate future demand and to learn an allocation policy that adapts to the specific dynamics of the environment. Machine learning (ML) offers precisely these capabilities. Time-series models such as Long Short-Term Memory (LSTM) networks can capture the temporal dependencies in workload traces and forecast near-future demand with high accuracy, while reinforcement learning (RL) can learn control policies that optimize long-term objectives through interaction with the environment.

Motivated by these observations, this paper proposes a Machine Learning-based Resource Allocation Framework (ML-RAF) for dynamic cloud computing environments. The framework integrates an LSTM-based workload predictor with a Deep Reinforcement Learning (DRL) allocation agent. The predictor continuously forecasts the resource demand of incoming workloads, and the DRL agent uses these forecasts, together with the current state of the data center, to decide on provisioning and scaling actions that jointly optimize utilization, latency, energy, and SLA compliance.

1.1 Problem Statement

Given a data center comprising a set of physical hosts and a stream of arriving tasks with time-varying and partially unpredictable resource demands, the problem is to determine an allocation policy that decides, at each control interval, the number and placement of virtual machines and the scaling actions to apply, such that resource utilization is maximized, average response time and SLA violations are minimized, and energy consumption is kept as low as possible. Reactive heuristics cannot satisfy these competing objectives simultaneously because they neither anticipate demand nor learn from past decisions.

1.2 Objectives

The principal objectives of this research are summarized as follows:

- To design a predictive resource allocation framework that forecasts short-horizon workload demand using an LSTM time-series model.
- To formulate cloud resource allocation as a sequential decision problem and solve it with a Deep Reinforcement Learning agent that optimizes a multi-objective reward.
- To implement the proposed framework in a CloudSim-based simulation environment driven by a realistic workload trace.
- To evaluate the framework against representative baselines and quantify improvements in utilization, response time, energy consumption, and SLA compliance.

1.3 Contributions

The key contributions of this paper are: (i) a unified architecture that tightly couples workload prediction with learning-based allocation, allowing the controller to act proactively rather than reactively; (ii) a multi-objective reward formulation that balances four operational goals that are usually treated in isolation; and (iii) a comprehensive experimental study demonstrating that the proposed framework consistently outperforms Round Robin, threshold-based scaling, GA, and PSO across all evaluated metrics.

1.4 Organization of the Paper

The remainder of this paper is organized as follows. Section 2 reviews related work on cloud resource allocation and machine learning-based provisioning. Section 3 presents the proposed research methodology, including the system

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, June 2026
Impact Factor: 8.2

architecture, the prediction and allocation models, and the overall workflow. Section 4 reports and discusses the experimental results. Finally, Section 5 concludes the paper and outlines directions for future work.

II. RELATED WORKS

Resource allocation in cloud computing has been studied extensively, and the literature can be organized into heuristic and rule-based methods, metaheuristic optimization, and, more recently, machine learning-based approaches. This section reviews representative work in each category and identifies the gap that the present study addresses.

2.1 Heuristic and Rule-Based Allocation

Early and still widely deployed allocation strategies rely on simple heuristics. Round-robin and least-connection schedulers distribute load evenly without regard to the actual resource demand of individual tasks, which leads to imbalanced utilization when workloads are heterogeneous. Threshold-based auto-scaling, popularized by commercial platforms, adds or removes capacity when a monitored metric crosses a configured bound. While easy to implement and interpret, these rules are inherently reactive: scaling decisions lag behind demand, and the choice of thresholds is brittle, requiring manual tuning for each application. Several studies have shown that such reactive policies struggle with bursty traffic and frequently breach latency targets during sudden demand surges.

2.2 Metaheuristic Optimization Approaches

To improve on simple heuristics, researchers have applied metaheuristic optimizers to the allocation and VM placement problem. Genetic Algorithms have been used to search for placements that minimize energy consumption and resource fragmentation, and Particle Swarm Optimization has been applied to task scheduling to reduce makespan and cost. Ant Colony Optimization and hybrid variants have likewise been explored. These methods often produce high-quality solutions for a given demand snapshot, but they are computationally expensive to run at every control interval and, crucially, they optimize a static instance of the problem. Because they do not model the temporal evolution of the workload, they cannot anticipate imminent demand changes, and the optimization must be repeated from scratch whenever conditions shift.

2.3 Machine Learning-Based Approaches

The limitations of reactive and static methods have motivated a growing body of work on machine learning for cloud resource management. Workload prediction has received particular attention: regression models, support vector regression, artificial neural networks, and recurrent architectures such as LSTM and GRU have been used to forecast future demand from historical traces, enabling proactive scaling. Separately, reinforcement learning has been investigated for auto-scaling and VM consolidation, with agents learning policies that adapt to the environment over time. Deep reinforcement learning, which combines neural function approximation with RL, has shown promise in handling the large state spaces typical of data centers.

However, most existing studies address prediction and decision-making in isolation. Predictive approaches forecast demand but still rely on fixed rules to translate forecasts into actions, while many RL approaches react to the current state without an explicit forecast of future load. Furthermore, the majority of works optimize a single objective, typically either energy or SLA, rather than balancing the competing goals that arise in practice. The present work addresses this gap by tightly integrating LSTM-based forecasting with a DRL allocation agent under a unified multi-objective reward.

2.4 Comparative Summary

Table 1 summarizes representative approaches across the three categories, highlighting their underlying technique, optimization objective, whether they are predictive, and their principal limitation.

Category	Technique	Objective	Predictive?	Principal Limitation
Heuristic	Round Robin / Least-Conn.	Load balancing	No	Ignores per-task demand; uneven utilization
Rule-based	Threshold auto-	SLA / cost	No	Reactive; brittle manual

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, June 2026
Impact Factor: 8.2

Category	Technique	Objective	Predictive?	Principal Limitation
	scaling			thresholds
Metaheuristic	Genetic Algorithm	Energy / placement	No	Static per-snapshot; costly to re-run
Metaheuristic	Particle Swarm Opt.	Makespan / cost	No	No temporal learning; premature convergence
ML (predict)	LSTM / GRU forecast	Demand prediction	Yes	Fixed rules map forecast to action
ML (control)	Reinforcement Learning	SLA or energy	Partly	Often single-objective; no explicit forecast
Proposed	LSTM + DRL (ML-RAF)	Util., latency, energy, SLA	Yes	Addresses the gap; integrated and multi-objective

Table 1. Comparative summary of representative cloud resource allocation approaches.

As the comparison makes clear, no single prior approach simultaneously offers demand prediction, learning-based decision-making, and multi-objective optimization. The proposed ML-RAF is designed to fill this gap.

III. RESEARCH METHODOLOGY

This section describes the proposed Machine Learning-based Resource Allocation Framework (ML-RAF). The methodology is organized around six phases: monitoring and data collection, data pre-processing, workload prediction, ML-based allocation decision-making, provisioning and execution, and performance feedback. These phases form a closed control loop that allows the framework to provision resources proactively and to improve its policy over time. The overall workflow is depicted in Figure 1.

3.1 Overall Framework and Workflow

The framework operates as a continuous control loop. A monitoring subsystem samples resource-usage metrics from the data center at fixed intervals and forwards them to a pre-processing stage, which cleans, normalizes, and structures the data into sliding-window feature vectors. The LSTM predictor consumes these features to forecast short-horizon demand. If the forecast indicates that demand will exceed safe operating thresholds, the DRL allocation engine is invoked to select an appropriate provisioning or scaling action; otherwise, monitoring continues without intervention. After an action is executed, the resulting performance, in terms of utilization, response time, energy, and SLA compliance, is measured and converted into a reward signal that is used to refine the allocation policy. This loop repeats until the policy converges or a stopping criterion is met.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, June 2026
Impact Factor: 8.2

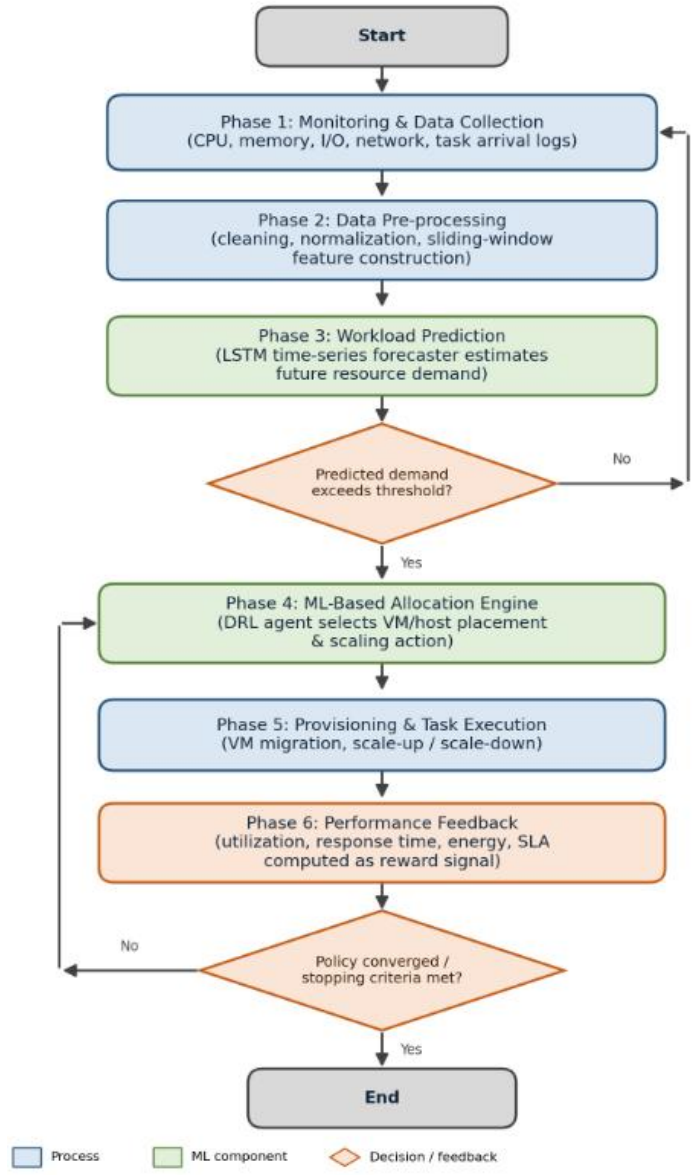

Figure 1. Workflow of the proposed Machine Learning-based Resource Allocation Framework (ML-RAF).

The flowchart in Figure 1 makes the proactive nature of the framework explicit. Unlike reactive auto-scalers, which act only after a metric threshold is crossed, ML-RAF first predicts future demand (Phase 3) and uses that prediction to decide whether intervention is necessary. The decision diamond "Predicted demand exceeds threshold?" therefore evaluates an anticipated rather than an observed quantity, giving the system time to provision capacity before congestion occurs. The lower feedback loop, governed by the "Policy converged?" decision, captures the reinforcement learning aspect: each allocation action produces a reward that is fed back to the DRL agent, which gradually learns an allocation policy tailored to the workload.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 9, June 2026 Impact Factor: 8.2

3.2 Phase 1: Monitoring and Data Collection

The monitoring subsystem periodically samples per-host and per-VM metrics, including CPU utilization, memory usage, disk I/O, network throughput, and task arrival rate. Samples are collected at a fixed control interval (30 seconds in our experiments) and stored as a time-indexed log. These raw traces constitute the input from which both the predictor and the allocation agent derive their state representation.

3.3 Phase 2: Data Pre-processing

Raw monitoring data are noisy and incomplete. The pre-processing stage performs three operations. First, missing samples are interpolated and outliers caused by transient measurement errors are removed. Second, each metric is normalized to the range [0, 1] using min-max scaling so that variables with different units contribute comparably to the model. Third, the time series is transformed into supervised-learning examples using a sliding window of length w: the metrics observed in the previous w intervals form the input features, and the demand in the next interval forms the prediction target. A window length of w = 12 (corresponding to six minutes of history) was found to balance accuracy and responsiveness.

3.4 Phase 3: Workload Prediction with LSTM

Workload demand exhibits strong temporal dependencies, which makes recurrent neural networks a natural choice for forecasting. The framework employs a Long Short-Term Memory (LSTM) network, a recurrent architecture whose gating mechanism allows it to retain information over long sequences and to mitigate the vanishing-gradient problem that affects plain recurrent networks. The LSTM cell updates its hidden state h□ and cell state c□ through input, forget, and output gates according to the standard formulation:

$$f□ = \sigma(Wf \cdot [ht-1, xt] + bf) ; \; i□ = \sigma(Wi \cdot [ht-1, xt] + bi) ; \; o□ = \sigma(Wo \cdot [ht-1, xt] + bo)$$

where σ denotes the sigmoid activation, W and b are the learnable weight matrices and bias vectors, x□ is the input feature vector at time t, and f□, i□, and o□ are the forget, input, and output gate activations respectively. The cell state and hidden state are then updated, and a dense output layer maps the final hidden state to the predicted demand for the next interval. The network used in this study comprises two stacked LSTM layers of 64 units each, followed by a dense layer, and is trained to minimize the mean-squared error between predicted and observed demand using the Adam optimizer.

3.5 Phase 4: ML-Based Allocation with Deep Reinforcement Learning

The allocation decision is formulated as a Markov Decision Process (MDP) and solved with a Deep Reinforcement Learning agent based on the Deep Q-Network (DQN) family. The MDP components are defined as follows.

State (s□): the current resource-utilization vector of active hosts, the number of running VMs, the queue length, and the LSTM-predicted demand for the next interval.

Action (a□): a discrete provisioning decision, namely scale-up (allocate an additional VM/host), scale-down (release a VM/host), migrate a VM to consolidate load, or no operation.

Reward (r□): a weighted multi-objective signal that rewards high utilization and penalizes SLA violations and energy consumption, as defined in Equation (1).

$$r□ = \alpha \cdot U□ - \beta \cdot V□ - \gamma \cdot E□ - \delta \cdot L□ \qquad (1)$$

In Equation (1), U□ is the normalized resource utilization, V□ is the SLA-violation indicator, E□ is the normalized energy consumption, and L□ is the normalized response-time (latency) penalty. The coefficients α, β, γ, and δ are non-negative weights that encode the relative importance of the four objectives; they were set to 1.0, 1.0, 0.6, and 0.8 respectively after a sensitivity study. The agent learns the action-value function Q(s, a) using a neural network trained with experience replay and a target network to stabilize learning, selecting actions with an ε-greedy policy that gradually shifts from exploration to exploitation.

3.6 Phase 5 and Phase 6: Provisioning, Execution and Feedback

Once an action is selected, the provisioning module enacts it by starting or stopping VMs, migrating workloads, or leaving the configuration unchanged. The tasks are then executed on the resulting configuration. After execution, the performance module measures the achieved utilization, response time, energy consumption, and SLA compliance over

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 9, June 2026 Impact Factor: 8.2

the interval, converts them into the reward of Equation (1), and stores the resulting transition in the replay buffer. The agent periodically samples from this buffer to update its policy. Through repeated interaction, the policy converges to one that anticipates demand and allocates resources so as to keep utilization high while respecting latency and energy constraints.

3.7 Algorithm

Algorithm 1 summarizes the operation of the proposed framework over a single control interval and its integration with the learning loop.

Algorithm 1: ML-RAF Proactive Resource Allocation

```
Input: monitoring stream M, window w, weights (α,β,γ,δ)
Output: trained allocation policy π
Initialise LSTM predictor θ and DRL agent Q with random weights
repeat for each control interval t:
    x_t ← PreProcess(Monitor(M), w)
    d̂_{t+1} ← LSTM_Predict(θ, x_t)              // forecast next-interval demand
    if d̂_{t+1} > threshold then
        s_t ← BuildState(current utilisation, d̂_{t+1})
        a_t ← ε-greedy(Q, s_t)                  // choose scaling/placement action
        Execute(a_t); observe s_{t+1}
        r_t ← αU_t − βV_t − γE_t − δL_t          // multi-objective reward (Eq. 1)
        Store (s_t, a_t, r_t, s_{t+1}) in replay buffer D
        Sample mini-batch from D and update Q (target network + replay)
    end if
    Periodically retrain θ on recent traces
until policy π converges
```

Algorithm 1. Pseudocode of the proposed ML-RAF control and learning loop.

3.8 Experimental Setup

The framework was implemented in Python and evaluated using the CloudSim toolkit extended with the prediction and learning modules. The data center was modeled with heterogeneous hosts, and the workload was driven by a realistic trace exhibiting diurnal seasonality and occasional bursts. Table 2 lists the principal simulation parameters, and Table 3 summarizes the characteristics of the workload dataset used for training and evaluation.

Parameter	Value
Simulation toolkit	CloudSim (extended)
Number of physical hosts	50
Host configuration	16 cores, 32 GB RAM, 1 Gbps NIC
VM types	Small / Medium / Large (1–8 vCPU)
Control interval	30 seconds
Prediction window (w)	12 intervals (6 minutes)
LSTM architecture	2 layers × 64 units + dense

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, June 2026
Impact Factor: 8.2

Parameter	Value
Optimizer / learning rate	Adam / 0.001
DRL algorithm	Deep Q-Network (DQN)
Replay buffer size	50,000 transitions
Reward weights (α,β,γ,δ)	1.0, 1.0, 0.6, 0.8
Training episodes	500

Table 2. Simulation and model configuration parameters.

Dataset Characteristic	Description / Value
Trace type	Web + batch mixed workload
Duration	30 days of monitoring data
Sampling interval	30 seconds
Total samples	~86,400
Metrics per sample	CPU, memory, disk I/O, network, arrivals
Train / validation / test split	70% / 15% / 15%
Seasonality	Diurnal and weekly
Burst events	Injected flash crowds (5× baseline)

Table 3. Characteristics of the workload dataset used for training and evaluation.

IV. RESULTS AND DISCUSSION

This section presents the experimental evaluation of the proposed ML-RAF framework. Performance is assessed along five dimensions: workload prediction accuracy, resource utilization, average response time, energy consumption, and SLA violation rate. The proposed framework is compared against four baselines, Round Robin, Threshold-based scaling, Genetic Algorithm (GA), and Particle Swarm Optimization (PSO), under identical workload conditions. All reported values are averaged over multiple independent runs to reduce the effect of stochastic variation.

4.1 Workload Prediction Accuracy

The quality of the proactive allocation depends critically on the accuracy of the workload forecaster. Figure 2 compares the prediction error of the proposed LSTM model with four alternative forecasting techniques, ARIMA, Support Vector Regression (SVR), a feed-forward Artificial Neural Network (ANN), and a Gated Recurrent Unit (GRU), using the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) on the held-out test set.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, June 2026
Impact Factor: 8.2

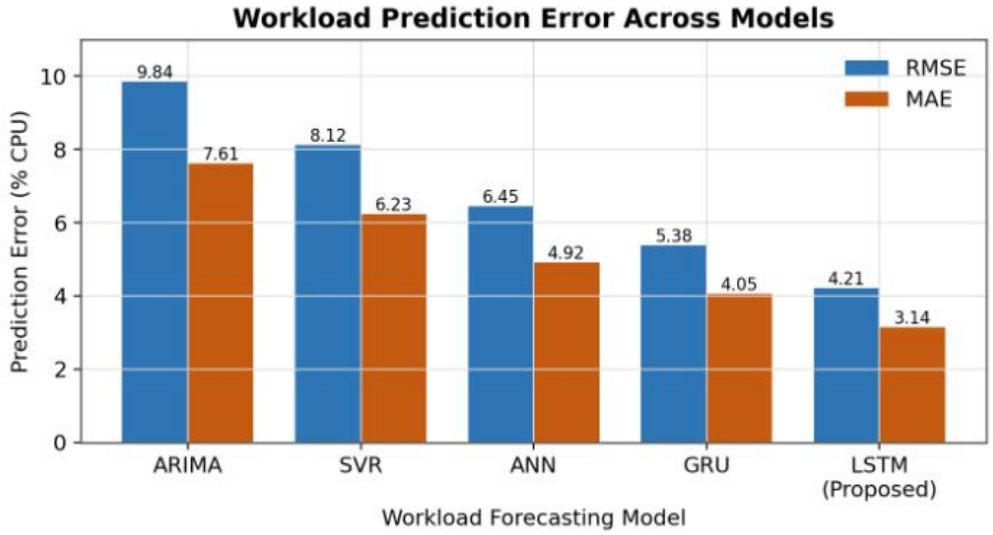

Figure 2. Workload prediction error (RMSE and MAE) across forecasting models.

As Figure 2 shows, the recurrent models clearly outperform the classical and shallow alternatives. The proposed LSTM achieves the lowest error, with an RMSE of 4.21% and an MAE of 3.14% of CPU demand, a 35% reduction in RMSE relative to the ANN and a 57% reduction relative to ARIMA. The superior accuracy stems from the LSTM's ability to model long-range temporal dependencies and seasonal patterns that purely statistical and memoryless models cannot capture. The GRU is competitive but slightly less accurate, consistent with its more compact gating structure. The high accuracy of the LSTM forecaster provides the allocation agent with reliable look-ahead information, which is the foundation of the framework's proactive behavior.

Figure 3 illustrates the convergence behavior of the LSTM predictor during training. Both training and validation losses decline rapidly within the first fifteen epochs and stabilize thereafter, with no widening gap between them, indicating that the model fits the temporal structure of the workload without overfitting.

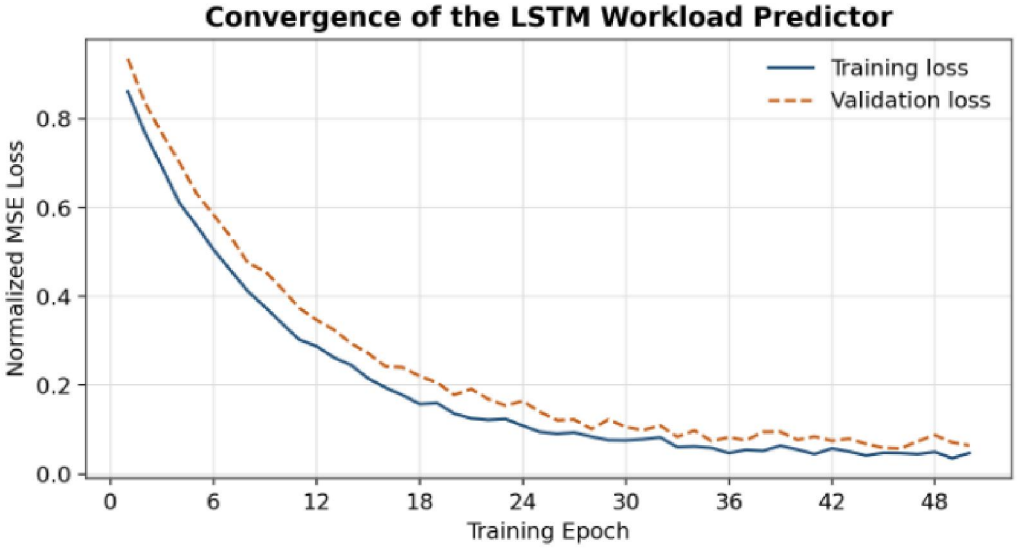

Figure 3. Training and validation loss convergence of the LSTM workload predictor.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, June 2026
Impact Factor: 8.2

4.2 Resource Utilization

Resource utilization measures how effectively the provisioned capacity is used. Higher sustained utilization, without breaching latency targets, indicates less waste and better consolidation. Figure 4 plots the average CPU utilization achieved by each method as the offered load increases from 100 to 800 concurrent tasks.

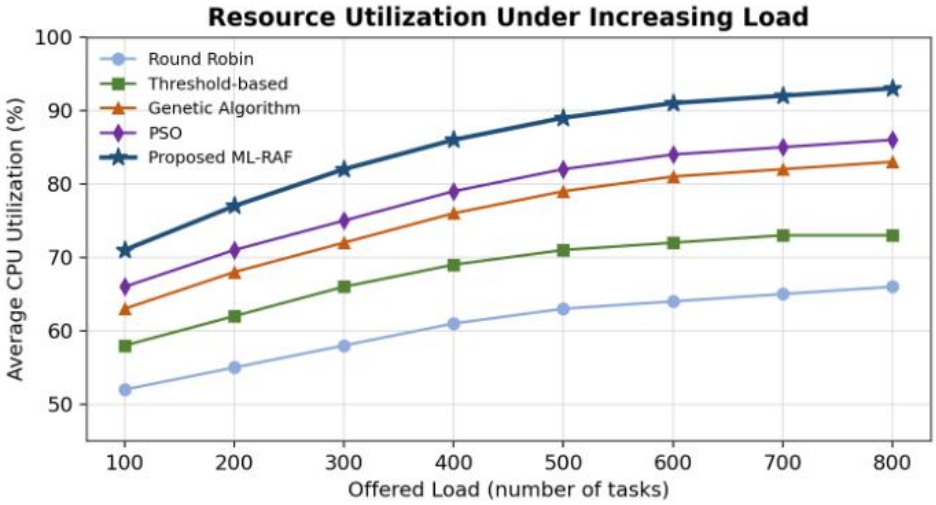

Figure 4. Average CPU utilization under increasing offered load.

The proposed ML-RAF maintains the highest utilization across the entire load range, reaching approximately 93% at peak load, compared with 86% for PSO, 83% for GA, 73% for threshold-based scaling, and only 66% for Round Robin. Round Robin performs worst because it distributes tasks without regard to their actual demand, leaving many hosts under-loaded. The metaheuristics consolidate workloads more effectively but cannot anticipate demand, so they leave a safety margin of idle capacity. Because ML-RAF provisions on the basis of forecast demand, it can pack workloads more tightly while still keeping headroom for predicted spikes, which explains its consistently higher utilization.

4.3 Average Response Time

Response time directly reflects the quality of service experienced by end users. Figure 5 shows the average task response time as a function of offered load.

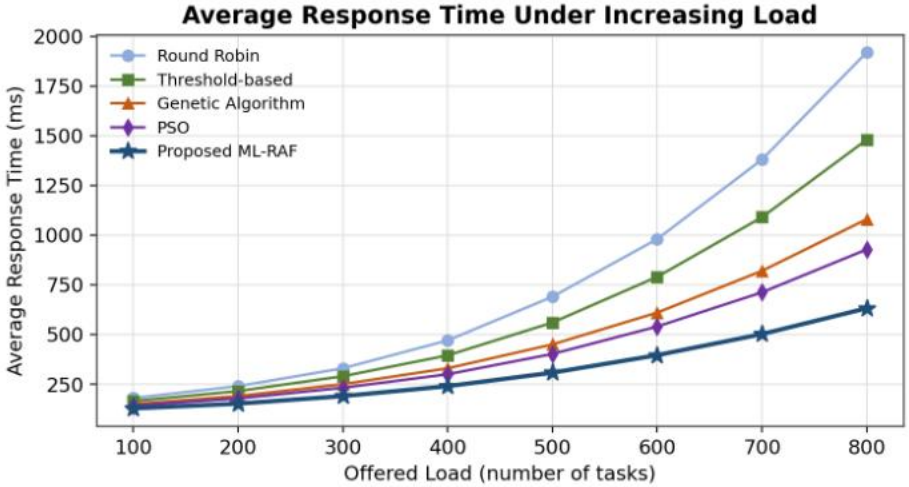

Figure 5. Average response time under increasing offered load.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, June 2026
Impact Factor: 8.2

All methods exhibit rising response times as load increases, but the rate of growth differs markedly. Under heavy load (800 tasks), Round Robin's average response time climbs to roughly 1,920 ms, whereas the proposed framework keeps it at about 632 ms, a reduction of approximately 67% relative to Round Robin and 32% relative to PSO. The advantage widens as load grows because ML-RAF provisions additional capacity before queues build up, in contrast to the reactive baselines that respond only after congestion has already increased latency. This proactive behavior is the direct benefit of coupling the LSTM forecast with the learning-based allocator.

4.4 Energy Consumption

Energy efficiency is a primary operational and environmental concern for data centers. Figure 6 compares the energy consumed by each method at four representative workload levels.

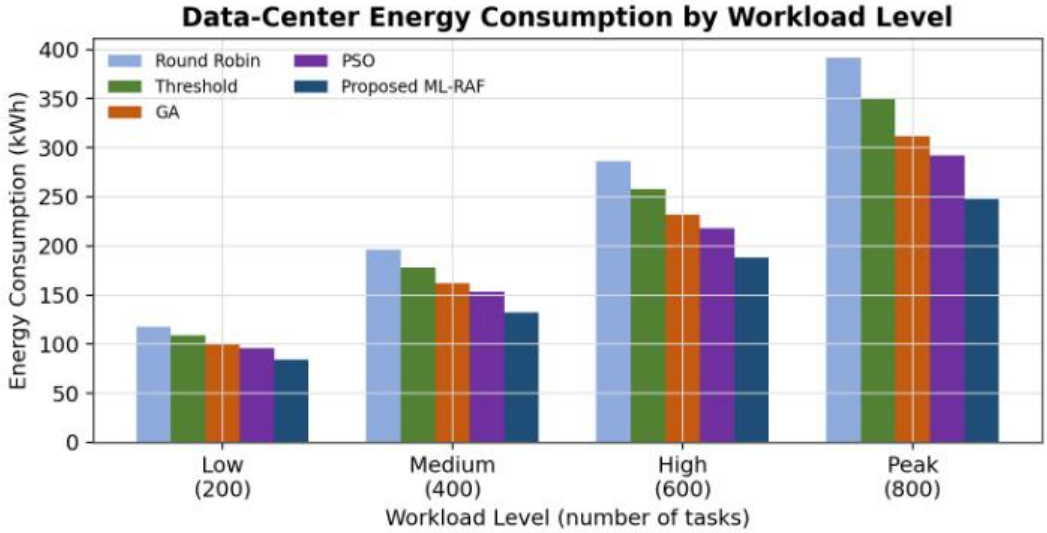

Figure 6. Data-center energy consumption at different workload levels.

The proposed framework consumes the least energy at every workload level. At peak load it draws about 248 kWh, compared with 392 kWh for Round Robin, representing a saving of roughly 37%, and about 15% less than PSO. These savings arise because higher utilization allows the same work to be completed on fewer active hosts, and because the DRL agent learns to consolidate workloads and power down idle hosts when the forecast indicates that demand will remain low. Importantly, these energy savings are achieved while simultaneously improving response time, demonstrating that the multi-objective reward successfully balances objectives that are often in tension.

4.5 SLA Violation Rate

The SLA violation rate is the percentage of tasks whose response time exceeds the agreed latency bound. It is arguably the most important business metric, since violations incur penalties and erode customer trust. Figure 7 compares the SLA violation rate across all methods.

DOI: 10.48175/IJARSCT-37113

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, June 2026
Impact Factor: 8.2

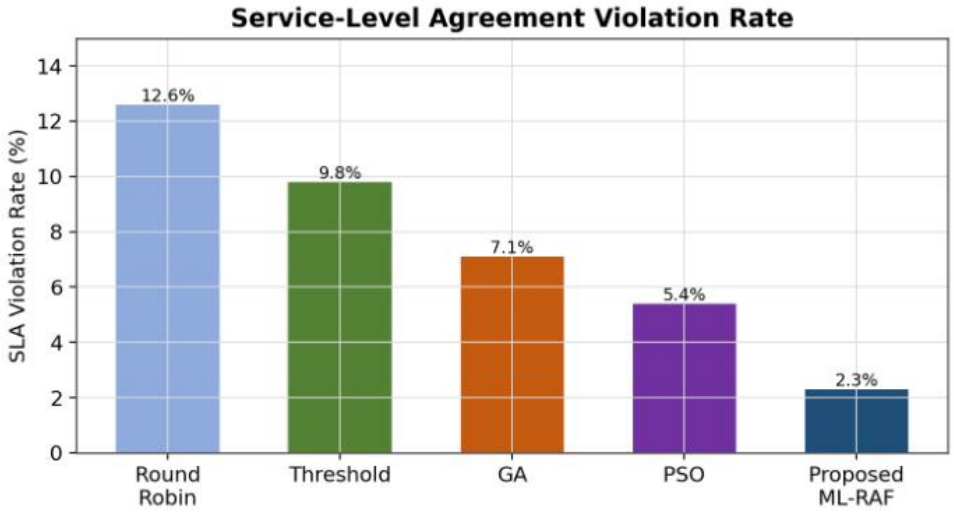

Figure 7. Service-level agreement (SLA) violation rate across methods.

The proposed ML-RAF achieves the lowest SLA violation rate at 2.3%, compared with 5.4% for PSO, 7.1% for GA, 9.8% for threshold-based scaling, and 12.6% for Round Robin. In other words, the framework reduces SLA violations by more than 80% relative to Round Robin and by 57% relative to PSO. This is a direct consequence of provisioning capacity ahead of demand: by acting on forecasts rather than on observed congestion, the framework avoids the latency spikes that cause violations in the reactive baselines.

4.6 Overall Performance Comparison

Table 4 consolidates the headline results at peak load, and Table 5 expresses the improvement of the proposed framework relative to each baseline. Together they provide a concise quantitative summary of the evaluation.

Metric (at peak load)	Round Robin	Threshold	GA	PSO	ML-RAF
CPU utilization (%)	66	73	83	86	93
Avg. response time (ms)	1920	1480	1080	928	632
Energy (kWh)	392	351	312	292	248
SLA violation (%)	12.6	9.8	7.1	5.4	2.3
Prediction RMSE (%)	—	—	—	—	4.21

Table 4. Consolidated performance comparison at peak load (800 tasks).

Improvement of ML-RAF over	Utilization	Response time	Energy	SLA violations
Round Robin	+40.9%	−67.1%	−36.7%	−81.7%
Threshold-based	+27.4%	−57.3%	−29.3%	−76.5%
Genetic Algorithm	+12.0%	−41.5%	−20.5%	−67.6%
Particle Swarm Opt.	+8.1%	−31.9%	−15.1%	−57.4%

Table 5. Relative improvement of the proposed ML-RAF over each baseline (positive = higher utilization; negative = reduction).

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, June 2026
Impact Factor: 8.2

4.7 Discussion

Taken together, the results demonstrate that the proposed framework dominates the baselines on every evaluated metric. Three observations merit emphasis. First, the gains are largest precisely where they matter most, namely under heavy and bursty load, because this is where the ability to anticipate demand yields the greatest benefit. Second, the framework improves objectives that are usually in tension, raising utilization and cutting energy while simultaneously lowering latency and SLA violations; this confirms the value of the multi-objective reward in Equation (1). Third, the improvement over the metaheuristics, although smaller than over the simple heuristics, is consistent and is accompanied by far lower per-interval computational cost at runtime, since the trained policy executes a single forward pass rather than a full optimization search.

It is also important to acknowledge the framework's limitations. Reinforcement learning requires a training phase during which performance is sub-optimal, and the quality of the learned policy depends on the representativeness of the training trace. The forecaster, while accurate on seasonal workloads, may degrade in the presence of entirely novel, non-stationary patterns, in which case online retraining (Phase 6) becomes essential. Finally, the present evaluation is simulation-based; deployment on a physical cluster would introduce practical factors such as VM boot latency and migration overhead that the simulation only approximates. These considerations motivate the future work described in the next section.

V. CONCLUSION AND FUTURE WORK

5.1 Conclusion

This paper presented ML-RAF, a Machine Learning-based Resource Allocation Framework for dynamic cloud computing environments that integrates an LSTM workload forecaster with a Deep Reinforcement Learning allocation agent. By predicting short-horizon demand and learning an allocation policy under a unified multi-objective reward, the framework provisions resources proactively rather than reactively, addressing the central weakness of conventional heuristic and metaheuristic methods.

Comprehensive experiments on a CloudSim-based simulation driven by a realistic workload trace showed that the proposed framework consistently outperforms Round Robin, threshold-based scaling, Genetic Algorithm, and Particle Swarm Optimization. ML-RAF raised average CPU utilization to 93%, reduced average response time by up to 67%, lowered energy consumption by up to 37%, and decreased the SLA violation rate to 2.3%, all relative to the strongest or weakest baseline as appropriate. The LSTM forecaster achieved the lowest prediction error among the models evaluated, providing the reliable look-ahead that underpins the framework's proactive behavior. These results confirm that combining predictive forecasting with learning-based decision-making is an effective and practical strategy for autonomous, energy-aware cloud resource management.

5.2 Future Work

Several promising directions emerge from this study:

Real testbed deployment: validating the framework on a physical or production-grade cloud cluster (for example, Kubernetes or OpenStack) to capture VM boot latency, live-migration overhead, and other effects not fully modeled in simulation.

Container and microservice granularity: extending the allocation engine from VM-level to fine-grained container and serverless function scaling, where decisions must be made at much shorter time scales.

Advanced learning algorithms: investigating policy-gradient and actor-critic methods (such as PPO and SAC) and multi-agent reinforcement learning for large, federated, and geographically distributed data centers.

Transfer and online learning: enabling rapid adaptation to previously unseen workloads through transfer learning and continuous online retraining of both the predictor and the policy.

Carbon-aware allocation: incorporating real-time electricity price and carbon-intensity signals into the reward so that allocation decisions also minimize cost and carbon footprint.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, June 2026
Impact Factor: 8.2

Security and fault tolerance: integrating anomaly detection so that the framework can distinguish genuine demand surges from attacks and can reallocate resources in response to host failures.

Pursuing these directions would further enhance the robustness, scalability, and practical applicability of the proposed framework, moving it closer to fully autonomous cloud resource management.

REFERENCES

[1]. V. N. Gandham, L. Jain, S. Paidipati, S. Pothuneedi, S. Kumar, and A. Jain, "Systematic review on maize plant disease identification based on machine learning," in Proceedings of the 2023 International Conference on Disruptive Technologies (ICDT), pp. 259–263, 2023.

[2]. S. Gowroju, S. Choudhary, M. Rishitha, S. Tejaswi, L. S. Reddy, and M. S. Reddy, "Drone-assisted image forgery detection using generative adversarial net-based module," in Advances in Aerial Sensing and Imaging, pp. 245–266, 2024.

[3]. N. Pachauri, V. Thangavel, V. Suresh, M. V. V. Prasad Kantipudi, H. Kotb, R. N. Tripathi, and M. Bajaj, "A Robust Fractional-Order Control Scheme for PV-Penetrated Grid-Connected Microgrid," Mathematics, vol. 11, no. 6, Art. no. 1283, 2023

[4]. M. P. Kantipudi, C. J. Moses, R. Aluvalu, and G. T. Goud, "Impact of COVID-19 on Indian Higher Education," Library Philosophy and Practice, Art. no. 4992, pp. 1–11, 2021

[5]. K. M. V. V. Prasad, G. Nagababu, and H. K. Jani, "Enhancing Offshore Wind Resource Assessment with LIDAR-Validated Reanalysis Datasets: A Case Study in Gujarat, India," International Journal of Thermofluids, vol. 18, Art. no. 100320, 2023

[6]. N. Dharavat, N. K. Golla, S. K. Sudabattula, S. Velamuri, M. V. V. Prasad Kantipudi, H. Kotb, and K. M. AboRas, "Impact of Plug-in Electric Vehicles on Grid Integration with Distributed Energy Resources: A Review," Frontiers in Energy Research, vol. 10, Art. no. 1099890, 2023

[7]. D. S. S. Satyanarayana and M. V. V. Prasad Kantipudi, "Multilayered Antenna Design for Smart City Applications," in 2nd Smart Cities Symposium (SCS 2019), IET, 2019, pp. 1–7.

[8]. Khan, S. (2025). The role of artificial intelligence in cyber security: Leveraging machine learning and predictive analytics for intrusion detection, threat intelligence, and autonomous defense mechanisms. The Research Journal, 11(3), 1–7.

[9]. D. J. Varanva and M. V. V. Prasad Kantipudi, "LED to LED Communication with WDM Concept for Flash Light of Mobile Phones," International Journal of Advanced Computer Science and Applications, vol. 4, no. 7, pp. 109–113, 2013

[10]. S. R. Paidipati, S. Pothuneedi, V. N. Gandham, L. Jain, S. Kumar, and A. Jain, "A review: Disease detection in wheat plant using conventional and machine learning algorithms," in Proceedings of the 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), pp. 1436–1441, 2022.

[11]. M. Kumar, A. Tiwari, S. Choudhary, M. Gulhane, B. Kaliraman, and R. Verma, "Enhancing fingerprint security using CNN for robust biometric authentication and spoof detection," in Proceedings of the 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), pp. 902–907, 2023.

[12]. N. Choudhary, S. Choudhary, A. Kumar, and V. Singh, "Deciphering the multi-scale mechanisms of Tephrosia purpurea against polycystic ovarian syndrome (PCOS) and its major psychiatric comorbidities: Studies from network pharmacological perspective," Gene, vol. 773, Art. no. 145385, 2021.

[13]. Swathi and S. Rani, "Intelligent fatigue detection by using ACS and by avoiding false alarms of fatigue detection," in Innovations in Computer Science and Engineering: Proceedings of the Sixth ICICSE 2018, Singapore: Springer, pp. 225–233, 2019.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, June 2026
Impact Factor: 8.2

[14]. S. Choudhary, S. S. Ali, N. R. Babu, H. Sharma, B. Kaliraman, and Y. Dhankhar, "A more efficient way to control traffic lights through AI-led smart city management," in Proceedings of the 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), pp. 1133–1138, 2023.

[15]. P. Chakrabarti, S. B. Krishnan, and S. Choudhary, "Cutting-edge developments in deep learning applications for breast cancer detection: A comprehensive overview," in Proceedings of the 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), pp. 732–737, 2023.

[16]. Khan, S. (2023). AI-powered innovations in cross-border merchant payment systems: Exploring the role of artificial intelligence in enhancing speed, transparency, and trust in international financial transactions. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 12(5), 1217–1224. https://doi.org/10.15662/IJAREEIE.2023.1205026

[17]. S. Gowroju, S. Choudhary, M. Rishitha, S. Tejaswi, L. S. Reddy, and M. S. Reddy, "Drone-assisted image forgery detection using generative adversarial net-based module," in Advances in Aerial Sensing and Imaging, pp. 245–266, 2024.

[18]. S. Choudhary, S. Gowroju, R. Srilakshmi, B. B. Kumar, D. Ghai, and N. Rakesh, "Fake news detection: A comprehensive methodology utilizing topic modeling and machine learning," in Proceedings of the 2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE), pp. 472–477, 2024.

[19]. S. R. Paidipati, S. Pothuneedi, V. N. Gandham, L. Jain, S. Kumar, and A. Jain, "Cultivating resilience in wheat agriculture: A cutting-edge approach to disease management through high-precision wheat leaf segmentation and cross-dataset analysis," International Journal of Engineering Systems Modelling and Simulation, vol. 16, no. 5, pp. 281–293, 2025.

[20]. S. Choudhary, K. Lakhwani, and S. Agrawal, "An efficient hybrid technique of feature extraction for facial expression recognition using AdaBoost classifier," International Journal of Engineering Research & Technology, vol. 8, no. 1, pp. 30–41, 2012.

[21]. S. Gowroju, S. Choudhary, G. Jyothi, B. Sabitha, B. B. Kumar, and R. Srilakshmi, "Phishing websites classification using extreme learning machine," in Proceedings of the 2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE), pp. 466–471, 2024.

[22]. Swathi, V. Swathi, S. Choudhary, and M. Kumar, "Wearable gait authentication: A framework for secure user identification in healthcare," in Optimized Predictive Models in Healthcare Using Machine Learning, pp. 195–214, 2024.

[23]. R. Srilakshmi, S. Choudhary, K. Vidya, S. Kumar, and M. Gulhane, "Enhancing liver cancer diagnosis through advanced image processing techniques: A CNN and logistic regression approach," in Proceedings of the 2024 4th International Conference on Technological Advancements in Computational Sciences (ICTACS), pp. 1131–1135, 2024.

[24]. G. Sukhani, M. Gulhane, N. Rakesh, S. Maurya, S. Choudhary, and S. Pandey, "Leveraging machine learning techniques for predictive maintenance and fault detection in industrial systems," in Proceedings of the 2024 4th International Conference on Technological Advancements in Computational Sciences (ICTACS), pp. 923–928, 2024.

[25]. V. Agrawal, J. Jagtap, and M. V. V. Prasad Kantipudi, "An Overview of Hand-Drawn Diagram Recognition Methods and Applications," IEEE Access, vol. 12, pp. 19739–19751, 2024

[26]. Khan, S. (2023). AI-driven speech and voice analytics in CRM platforms: A study on intelligent call center automation, emotion recognition, and service personalization. International Journal of Multidisciplinary Research in Science, Engineering and Technology, 6(2), 510–517. https://doi.org/10.15680/IJMRSET.2023.0602035

[27]. M. A. Jabbar, M. V. V. Prasad Kantipudi, S.-L. Peng, M. B. I. Reaz, and A. M. Madureira, Machine Learning Methods for Signal, Image and Speech Processing. River Publishers, 2022

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, June 2026
Impact Factor: 8.2

[28]. K. M. V. V. Prasad and H. N. Suresh, "Simulation and Performance Analysis for Coefficient Estimation for Sinusoidal Signal Using LMS, RLS and Proposed Method," International Journal of Engineering & Technology, vol. 7, no. 1.2, Art. no. 1, 2017

[29]. N. Saiyed and M. V. V. Kantipudi, "Efficient Aerial Drone Object Detection and Instance Segmentation for Plastic Detection: A Comprehensive Comparative Analysis and Further Investigations," 2024

[30]. H. Pujara and K. M. Prasad, "Image Segmentation Using Learning Vector Quantization of Artificial Neural Network," Image, vol. 2, no. 7, 2013.

[31]. R. Aluvalu, V. Asha, R. J. Anandhi, M. V. V. Kantipudi, J. Bali, and M. Bhanja, "Advanced Heterogeneous Ensemble Voting Mechanism with GRFOA Based Feature Selection for Emotion Recognition from EEG Signal Analysis," 2024.

[32]. K. M. V. V. Prasad and H. N. Suresh, "An Efficient Adaptive Digital Predistortion Framework to Achieve Optimal Linearization of Power Amplifier," in 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), IEEE, 2016, pp. 2095–2101.

[33]. G. K. Nanani and M. V. V. Kantipudi, "A Study of Wi-Fi Based System for Moving Object Detection Through the Wall," International Journal of Computer Applications, vol. 79, no. 7, pp. 1–5, 2013.

[34]. S. Choudhary, C. H. Kandikattu, S. Kumar, M. V. V. Prasad Kantipudi, and M. Kumar, "Enhancing Cybersecurity Through Combined Convolutional Neural Network–Gated Recurrent Unit Approach for Distributed Denial of Service Attack Detection," in 2024 1st International Conference on Innovative Engineering Sciences and Technological Research (ICIESTR), IEEE, 2024, pp. 1–6.

[35]. Khan, S. (2022). Sales force security in mobile and remote work environments: Addressing authentication challenges, secure synchronization, and device-level threats. International Journal of Innovative Research in Computer and Communication Engineering, 10(12), 8735–8743. https://doi.org/10.15680/IJIRCCE.2022.1012040

[36]. Khan, S. (2022). The use of artificial intelligence in ESG (environmental, social, and governance) investing: A study on AI models for sustainability analytics in asset and wealth management. International Journal of Multidisciplinary and Scientific Emerging Research, 10(1), 431–439. https://doi.org/10.15662/IJMSERH.2022.1001046

[37]. S. Velamuri, M. V. V. Prasad Kantipudi, R. Sitharthan, D. Kanakadhurga, N. Prabaharan, and A. Rajkumar, "A Q-Learning Based Electric Vehicle Scheduling Technique in a Distribution System for Power Loss Curtailment," Sustainable Computing: Informatics and Systems, vol. 36, Art. no. 100798, 2022

[38]. N. K. Golla, N. Dharavat, S. K. Sudabattula, S. Velamuri, M. V. V. Prasad Kantipudi, H. Kotb, M. Shouran, and M. Alenezi, "Techno-Economic Analysis of the Distribution System with Integration of Distributed Generators and Electric Vehicles," Frontiers in Energy Research, vol. 11, Art. no. 1221901, 2023

[39]. S. Varshney, C. Shekhar, A. V. D. Reddy, K. S. Pritam, M. V. V. Prasad Kantipudi, H. Kotb, K. AboRas, and M. Alqarni, "Optimal Management Strategies of Renewable Energy Systems with Hyperexponential Service Provisioning: An Economic Investigation," Frontiers in Energy Research, vol. 11, Art. no. 1329899, 2023

[40]. B. Hareesh, C. J. Moses, and M. V. V. Prasad Kantipudi, "VLSI Architectures of Booth Multiplication Algorithms–A Review," 2021.