Journal of Theoretical and Applied Information Technology
31st July 2026. Vol.104. No.14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ENHANCING CLOUD COMPUTING EFFICIENCY WITH HYBRID MACHINE LEARNING ALGORITHMS FOR VM FAILURE DETECTION, RESCHEDULING, AND QOS MANAGEMENT

B Subramanya Anil Kumar[1], Dr Basant Sah[2]
Research Scholar, Department of CSE, *Koneru Lakshmaiah Education Foundation,* Guntur, AP
bsanilkumar2104@gmail.com

Associate Professor, Department of CSE, *Koneru Lakshmaiah Education Foundation,* Guntur, AP
basantbitmtech2008@kluniversity.in

ABSTRACT

This paper introduces an innovative framework aimed at optimizing cloud computing environments by incorporating Virtual Machine (VM) Failure Detection, Dynamic Task Rescheduling, and Quality of Service (QoS)-Based Performance Measurement. The proposed framework applies a combination of supervised machine learning models, including Linear Regression, KNN, SVR, boost, Light, and Random Forest, to predict key performance metrics such as CPU usage, memory usage, and network traffic. The integration of these models, particularly the combined Light and Random Forest, resulted in the most accurate predictions, achieving the lowest RMSE values across all metrics. The dataset used for the study contained essential cloud metrics like CPU and memory usage, network traffic, energy consumption, and task execution details. Through this framework, hybrid machine learning techniques are leveraged to detect VM failures in real-time while dynamically rescheduling tasks, optimizing resource distribution, and minimizing system downtime. The results demonstrate a marked improvement in cloud performance, with enhanced energy efficiency and reduced execution time. The experimental outcomes also showed significant improvements in failure detection accuracy and QoS metrics, such as reliability and security, making the framework highly effective for managing modern cloud systems. This work sets a new benchmark for cloud service performance by combining machine learning with AI-driven optimization techniques.

Keywords: *Hybrid Machine Learning, VM Failure Detection, Dynamic Task Rescheduling, Quality of Service (QoS), Cloud Computing.*

1. INTRODUCTION

Cloud computing gives consumers infinite computer resources. Businesses and consumers pay for these resources. These services offer elastic resource allocation, flexibility, horizontal and vertical scalability, and high availability. Kumar and Sharma benefit from consolidated cloud infrastructure management [1]. Virtualization and dynamic resource allocation improve user and provider resource consumption and waste. Effective scheduling ensures completion and resource use. Task scheduling is essential for cloud data centers as demand rises. Poor management can drain resources; thus, scheduling is vital. The schedule matches resources to activities. This boosts resource efficiency, task rejection, system reliability, energy efficiency, and operating expenses. Meeting SLAs. Task scheduling cuts resource waste. Task scheduling and resource delivery are cloud resource management strategies. Group Madni [2]. Resource scheduling includes mapping, brokering, load balancing, and allocation. First, we assess resource provisioning, then scheduling. Performance measures have pros and downsides for heuristics, meta-heuristics, and hybrid algorithms. Big data's dynamic nature helps cloud computing find VM faults, reorganize duties, and manage QoS. Optimizing complicated data cloud infrastructures requires these upgrades [3]. Large amounts of unstructured data harm clouds. Powerful real-time and batch data centers handle these volumes [4]. Enhanced virtual machine failure detection and rescheduling for service continuity. Big data foundations are taught using "5V" paradigm. Volume, velocity, variety, truth, and value matter. Clouds must generate real-time data quickly [5]. Social media updates and transaction data differ. Verified data prevents system failure from misleading data. Data helps firms improve cloud services and make decisions [6]. Hybrid machine learning matters in cloud. They predict virtual machine failures and reorganize workloads to improve efficiency. These

methods increase cloud system QoS such resource efficiency, job rejection rate, energy usage, and operational costs while meeting SLAs [7]. Machine learning models monitor virtual machines to prevent system downtime and ensure job execution as workloads climb. Data generation rises internationally. The 2020 data estimate is 40,000 Exabytes, 300 times 2005 [8]. Sector-wide data-intensive app adoption fosters this. Social media generates terabytes and internet transactions approach 450 billion everyday. Tracking user preferences helps businesses enhance marketing and product development [9]. Cloud MapReduce processes data live. This paradigm allows continuous processing of large datasets, ensuring cloud system reliability and cost-efficiency. These solutions improve VM failure management, workload rescheduling, and cloud service management when combined with hybrid machine learning [10]. Below figure 1 Illustrates large data communication, networking, processing infrastructure, and applications.

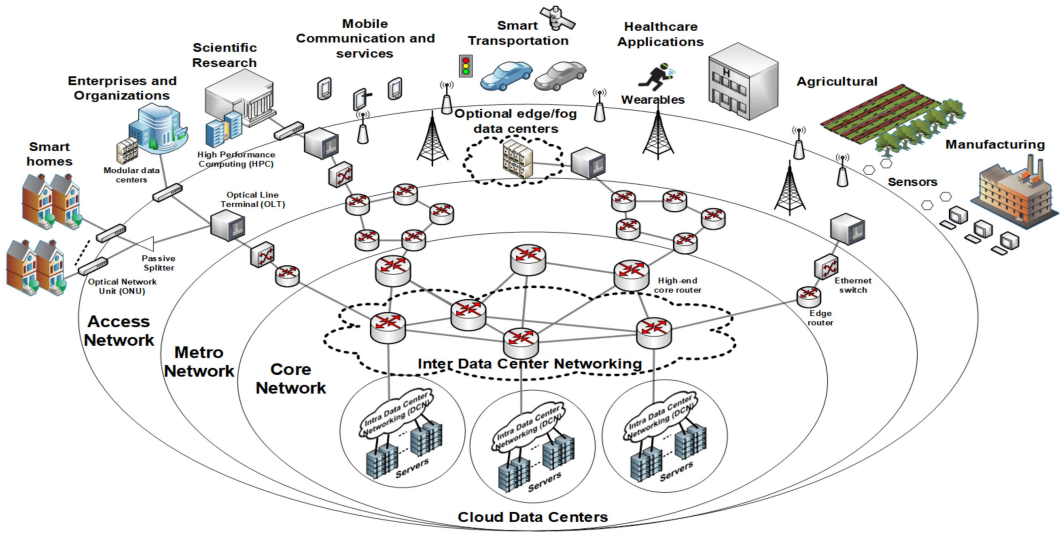

Figure 1: Illustrates large data communication, networking, processing infrastructure, and applications.

Cloud computing job scheduling is complex, especially when assigning interdependent tasks to VMs while meeting QoS constraints. S. Kaur et al [11]. Infrastructure as a Service (IaaS) or cloud computing makes these resources available online when needed. Shyam and Manvi [12]. Hybrid machine learning in cloud computing improves virtual machine fault diagnostics, rescheduling, and service management. Cloud computing has grown across academics, industry, and society due to internet use. This extension lets service providers add functional and non-functional aspects. Online resources, storage, and platforms are faster, more flexible, and cheaper with cloud computing. Amazon, Google, and Salesforce offer cloud services for these reasons. Zhang Group [14]. Virtualized cloud computing delivers apps online. System performance improves with cloud QoS. Maintaining CPU, bandwidth, and memory improves QoS. Hybrid machine learning and real-time variable adaptation reduce service delays and boost efficiency [15]. VM failure prediction and control reduce project downtime. Technology-driven scheduling and execution-time resource assignment maintain service quality. Team Demchenko. Many technologies improve performance, but none is best [16]. A load scheduler is needed in the cloud to algorithmically order jobs. Modified Round Robin (RR), a popular scheduling method, can be improved via machine learning to reduce waiting time and reaction time. A hybrid scheduling approach improves cloud computing performance. VM failure detection and workload rescheduling utilizing machine learning improve cloud QoS, resource efficiency, and QoS [17]. Below figure 2 shows Various resource management.

Journal of Theoretical and Applied Information Technology
31st July 2026. Vol.104. No.14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

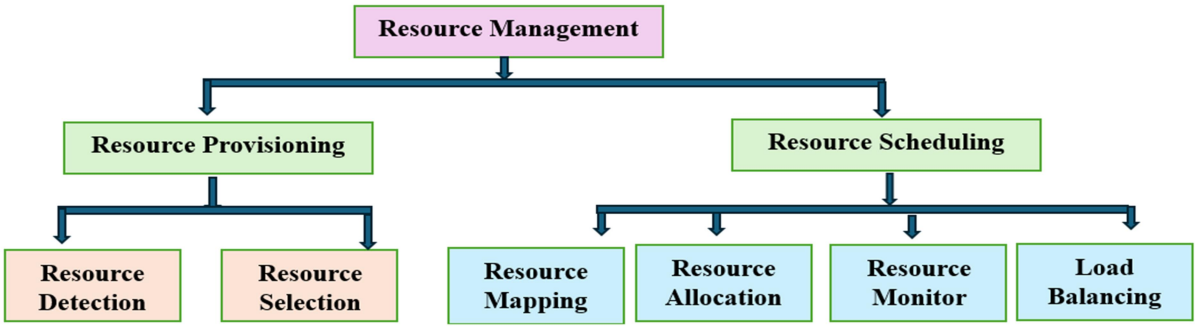

Figure 2: Various resource management.

Cloud computing changes local and global systems. With internet traffic and apps, cloud technologies developed. As grid computing and parallel processing grow, cloud computing lets users pay as they use [18]. This enables users to use dynamic, scalable virtual resources online. This strategy provides cheap, reliable services to startups. Cloud computing meets many computing demands due to its speed, flexibility, and adaptability. Data processing like labor scheduling and resource allocation increasingly uses hybrid machine learning [19]. These strategies improve job scheduling by assigning workloads to suitable VMs while maintaining QoS. Workflow and job scheduling increase cloud resource usage and fault tolerance. Cloud system reliability is considerably improved by hybrid machine learning. These methods group VM tasks by environment [20]. A proactive approach ensures cloud systems can withstand workload changes. Fault-tolerant scheduling evenly distributes jobs across nodes to reduce virtual machine failures [21]. Due to shifting cloud resources, cloud system maintenance requires fault-tolerant solutions. Excess clouds use damages parts. ML improves fail-tolerant systems [22]. For reliability and availability, cloud apps and services need fault tolerance. Memory overload, power outages, server failures, and VM defects cause task failures. By using hybrid machine learning, errors can be avoided [23]. Below figure 3 shows Resource allocation.

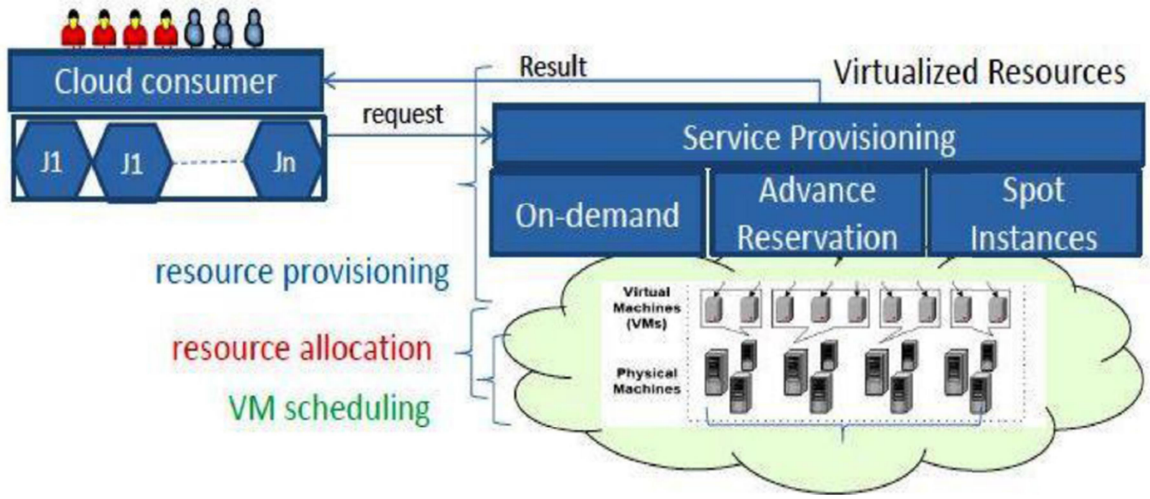

Figure 3: Resource allocation.

Checkpoints and redundancy accelerate and accept hybrid machine learning failures. Checkpoints work slowly. Multitasking and redundant hardware handle errors, delays, and system responsiveness [24]. These solutions take longer and consume more resources to balance performance and resource utilization in IaaS models, where speed and QoS are critical. VM failure detection, workload rescheduling, and QoS management are improved using hybrid machine learning in cloud computing. These solutions make cloud systems resilient and performant despite workload

variations and breakdowns [25]. Resource scheduling article identification reviewing technique was shown in figure 4.

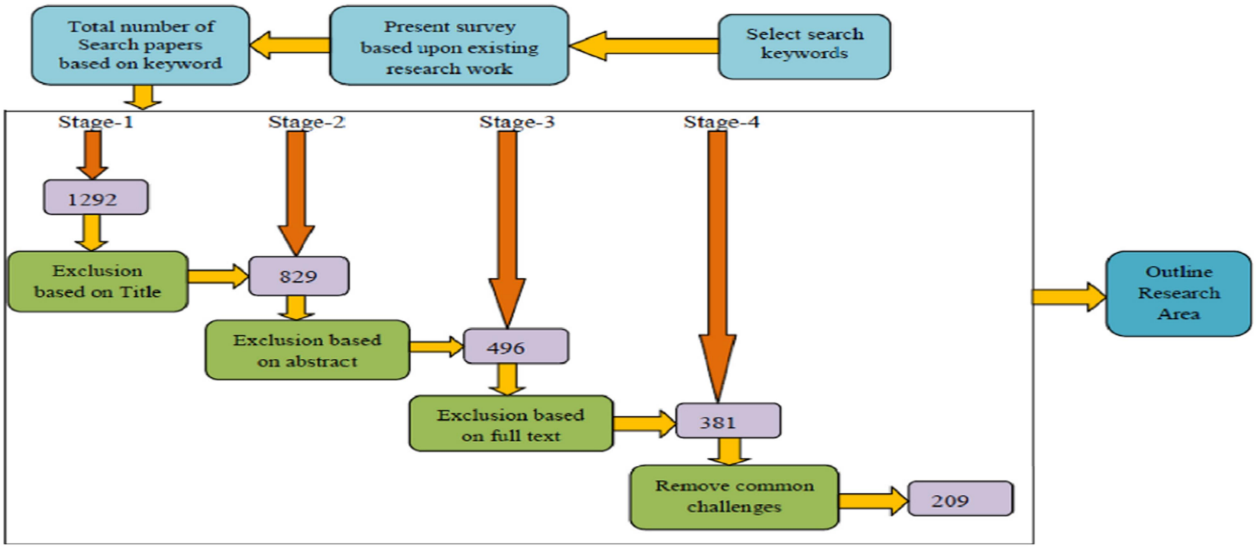

Figure 4: Resource scheduling article identification reviewing technique

1.2 Research Gap Statement

The literature reveals that no existing approach provides an integrated hybrid machine learning framework that jointly performs real-time VM failure detection, dynamic task rescheduling, and QoS-aware resource optimization while ensuring energy efficiency and SLA compliance in highly dynamic cloud environments.

This single sentence is extremely important because reviewers specifically look for the research gap.

1.3 Research Objectives

The primary objective of this study is to develop a hybrid machine learning framework for improving cloud computing efficiency through intelligent VM failure detection, dynamic task rescheduling, and QoS management.

The specific objectives are:

RO1: To develop a hybrid machine learning-based framework for accurate prediction of virtual machine failures in cloud environments.

RO2: To design a dynamic task rescheduling mechanism that minimizes system downtime and improves resource utilization.

RO3: To optimize Quality of Service parameters including reliability, availability, execution time, and energy consumption.

RO4: To evaluate the effectiveness of the proposed framework using cloud performance metrics such as CPU usage, memory utilization, and network traffic.

RO5: To compare the proposed hybrid framework with existing machine learning approaches and demonstrate improvements in prediction accuracy and cloud service performance.

1.4 Research Hypotheses

- **H1:** The proposed hybrid machine learning framework significantly improves VM failure detection accuracy.
- **H2:** Dynamic task rescheduling significantly reduces system downtime and task execution time.
- **H3:** The proposed framework significantly improves QoS metrics and SLA compliance.
- **H4:** The proposed framework significantly enhances resource utilization and energy efficiency.
- **H5:** The hybrid model significantly outperforms individual machine learning models in cloud performance prediction.

2. RELATED WORK

In cloud computing, load balancing, resource provisioning, application scheduling, and energy utilization are understudied. VM failure detection, workload rescheduling, and QoS resource allocation increase with hybrid machine

learning. The context will boost resource allocation and scheduling research. A detailed survey by M. Amiri and L. Khan [26] investigated prediction models for cloud resource provisioning and demonstrated their importance in improving allocation efficiency. However, the study primarily focused on workload prediction and did not consider real-time VM failure detection, adaptive task rescheduling, or QoS-aware optimization, thereby limiting its applicability in dynamic cloud environments. Their study covers resource provisioning, not over- and under-provisioning, which are crucial for VM management. J. Zhang et al. [27] investigated resource allocation and virtual machine migration strategies to improve cloud resource utilization and availability. Nevertheless, the proposed methods lack intelligent failure prediction mechanisms and do not incorporate machine learning-based QoS optimization, making them less suitable for highly dynamic cloud infrastructures. Both articles neglect machine learning's provisioning benefits. S. Smanchat and K. Viriyapant [28] presented a taxonomy of workflow scheduling techniques and provided a comprehensive classification of scheduling strategies in cloud computing. However, their study remained conceptual and did not evaluate intelligent learning-based approaches for handling VM failures and dynamic workload variations. They must study resource scheduling strategies to see if machine learning can help. To manage rescheduling and VM failure, P. Dave et al. [29] employed hybrid machine learning to compare scheduling options utilizing QoS criteria such resource consumption, dependability, scalability, and execution cost K. Radha et al. [30] matched requests to resources. They proposed a multi-tier capacity allocation system that optimizes real-time VM failure resource distribution using machine learning.

K. Ranjithprabhu and C. Nandhakumar [31] showed that heuristic and meta-heuristic resource scheduling strategies waste energy. Energy and task scheduling can benefit from machine learning. According to M. Kalra and S. Singh, five grid and cloud meta-heuristic scheduling algorithms employing Pareto optimal theory solved multi-objective issues such VM failure detection and QoS management [32]. According to their exhaustive review, machine learning could improve meta-heuristic methods for scheduling flexibility and fault tolerance. M. Masdari et al. [33] solved NP-Complete process scheduling with PSO. Their multi-objective and learning PSO analysis improves cloud computing VM failure management and rescheduling time, cost, and reliability. investigated meta-heuristics and static and dynamic resource allocation in 91 scheduling algorithm experiments [34]. The QoS-based allocation approaches they study build machine learning algorithms for resource scheduling to improve cloud efficiency. Cloud resource brokers let providers and clients port and interoperate. S. Chauhan et al. [35] examined resource broker strategies, their advantages and cons, and hybrid machine learning algorithm research.Finally, L. Bittencourt et al. [36] examined grid and cloud computing scheduling. Hybrid machine learning may increase cloud performance, virtual machine failure detection, and resource rescheduling. Finally, hybrid machine learning optimizes cloud computing VM failure management, workload rescheduling, and QoS [37]. Cloud systems increase resilience, energy efficiency, and performance by using machine learning for resource provisioning and scheduling. Table 1 shows each study's VM failure detection, rescheduling, and QoS management approaches and restrictions for cloud computing efficiency.

Table 1: Briefly describes each study's methods and limitations.

Author(s)	Study Focus	Methodology	Limitations
Yang and Chen[38]	Container scheduling to reduce resource fragmentation	Combined Gray Model (GM) and LSTM to predict resource usage, rescheduled to reduce idle resources	Focused mainly on resource fragmentation; other performance factors not deeply explored

Liu et al.[39]	Energy-efficient container scheduling	Linear regression model predicted physical machine (PM) resource usage, optimized scheduling, migration	Did not explore other AI techniques for predicting resource usage
Mehta et al.[40]	Power cap violation minimization in servers	Neural network-based model predicted power consumption based on CPU, memory, network, and disk usage	Primarily focused on power capping; lacks consideration of broader resource management issues
Tuli et al.[41]	Energy-efficient hybrid cloud resource management	Gated Graph Convolutional Network (GGCN) predicted QoS based on host temperature and scheduling correlations	Limited to energy cost and QoS; does not address full system optimization across all performance areas
Carvalho and Macedo [42]	QoE-aware container scheduling	LSTM and GRU networks predicted future QoE, ensured service-level objectives (SLO)	Limited to QoE metrics, without covering other operational factors like energy efficiency
Calheiros et al.[43]	Dynamic resource provisioning	ARIMA used to predict future workload, adjusted VM allocations based on predicted workload	Focused mainly on VM workload prediction; lacks exploration of complex, dynamic workloads
Bi et al. [44]	Large-scale workload prediction with high accuracy	Hybrid model combining variational mode decomposition, Bi-LSTM, grid LSTM, and attention mechanism	Primarily focused on prediction accuracy; does not address broader QoS management or resource scheduling
Xie et al. [45]	Container resource load prediction	Hybrid model using ARIMA and triple exponential smoothing for load sequence prediction	Concentrates on load prediction; broader VM scheduling strategies not discussed

Cloud computing QoS-aware autonomic resource management research is diverse. The goal is to create self-managing systems that can heal, configure, optimize, and protect.

2.1. Present QoS-aware Techniques for managing autonomous resources

Autonomous cloud computing systems diagnose and fix problems. SH-SLA systems prevent and correct SLA violations proactively and reactively. These solutions generally

underoptimize cloud service execution speed and cost [46].

Auto-configuring: Eco-adaptable systems. To meet SLAs, CBR and CoTuner use dynamic rules and policies to allocate resources in real time. This method frequently ignores cost and energy efficiency QoS. Performance is improved by self-optimizing systems' resource utilization and operational efficiency. AWM and CAS value low-cost, on-time work. Lack of heterogeneous cloud application support limits their use on multiple operating systems.

Self-protecting: Prevents system attacks. Secure Autonomic Technique (SAT) and Architecture Based Self-Protection (ABSP) detect and mitigate security threats in real time. These solutions are essential for cloud integrity and trust yet attack detection accuracy is generally poor.

2.2. Proposed Approach and Comparisons

Unlike others, our study's method has major disadvantages. Few systems have all four self-managing features. One framework with self-healing, self-configuring, self-optimizing, and self-protecting properties can meet several QoS criteria [47].

Research Contributions

Our model draws on previous studies but includes many major innovations

- **Integrated Approach:** Our paradigm unifies all four essential areas of self-management, improving cloud services holistically.
- **Enhanced QoS Management**: Our approach dynamically and intelligently controls resources to maximize cost, time, energy efficiency, and SLA adherence better than current models using hybrid machine learning techniques.
- **Real-Time Adaptability:** Our system's real-time adaptation to workload and environment changes makes cloud computing more durable and reliable.

2.3 Critical Literature Synthesis and Research Gap

The literature demonstrates significant progress in cloud resource provisioning, scheduling, and energy optimization. However, most existing studies address these challenges independently and fail to provide an integrated framework that simultaneously performs VM failure prediction, dynamic task rescheduling, and QoS-aware resource management. Furthermore, many approaches rely on single machine learning models, which often struggle to adapt to highly dynamic cloud environments and complex workload patterns. Therefore, there remains a research gap in developing an intelligent hybrid machine learning framework capable of achieving accurate failure prediction, adaptive rescheduling, and efficient QoS management in cloud computing environments. Motivated by these limitations, this study proposes a hybrid machine learning framework that integrates VM failure detection, dynamic task rescheduling, and QoS-aware resource management within a unified cloud computing architecture.

2.4 Problem Statement

Despite significant advancements in cloud computing, efficient management of Virtual Machine (VM) failures, dynamic task rescheduling, and Quality of Service (QoS) assurance remains a major challenge. Existing resource scheduling and fault-tolerance approaches primarily focus on individual objectives such as workload prediction, energy efficiency, or resource provisioning, while failing to provide an integrated framework that simultaneously addresses VM failure prediction, adaptive rescheduling, and QoS optimization. Recent studies have demonstrated the effectiveness of machine learning techniques for workload prediction and resource management; however, most existing methods suffer from limited real-time adaptability, inadequate fault tolerance, and poor handling of dynamic cloud environments. Furthermore, current approaches generally rely on single machine learning models, which often fail to capture the complex and non-linear relationships among cloud resource parameters, leading to increased system downtime, SLA violations, and inefficient resource utilization. Therefore, there is a critical need for an intelligent hybrid machine learning framework capable of accurately detecting VM failures, dynamically rescheduling tasks, and maintaining QoS requirements while improving energy efficiency and overall cloud performance.

2.5 Research Questions

RQ1: How can hybrid machine learning techniques improve the accuracy of VM failure prediction in cloud computing environments?

RQ2: How can dynamic task rescheduling reduce resource wastage and system downtime after VM failures?

RQ3: To what extent can the proposed framework improve QoS metrics such as reliability, availability, and energy efficiency?

RQ4: Does integrating multiple machine learning models provide better performance than individual models for cloud resource management?

RQ5: How effectively can the proposed framework adapt to dynamic cloud workloads while maintaining SLA requirements?

3. METHODOLOGY

A cloud computing system using hybrid machine learning methods to detect VM faults, dynamic workload rescheduling, and QoS management is recommended [30]. The hybrid method uses supervised and unsupervised learning models to improve resource allocation and service reliability and it was shown in figure 5.

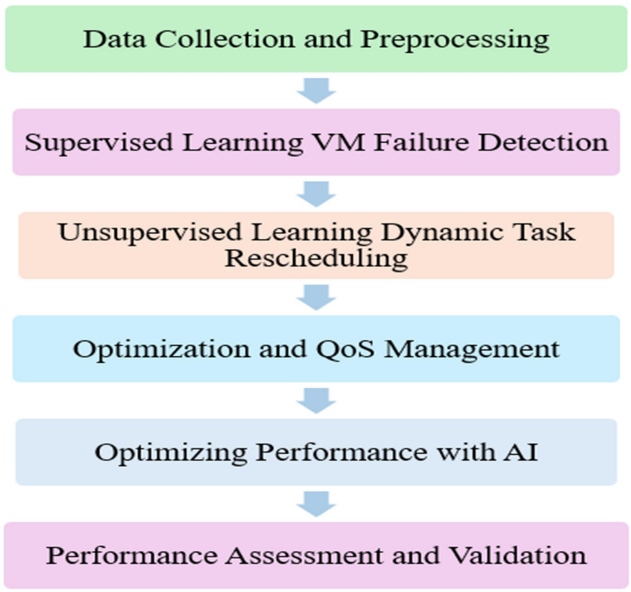

Figure 5: Workflow Methodology

3.1. CHOPPER: QoS-aware Autonomic Resource Management Approach

CHOPPER Architecture

CHOPPER is a full framework that auto-manages resources to meet SLA QoS requirements for cloud computing efficiency. Figure 6 depicts CHOPPER's design and functionality, focusing on dynamic cloud resource management without SLA violations [48].

Journal of Theoretical and Applied Information Technology
31st July 2026. Vol.104. No.14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

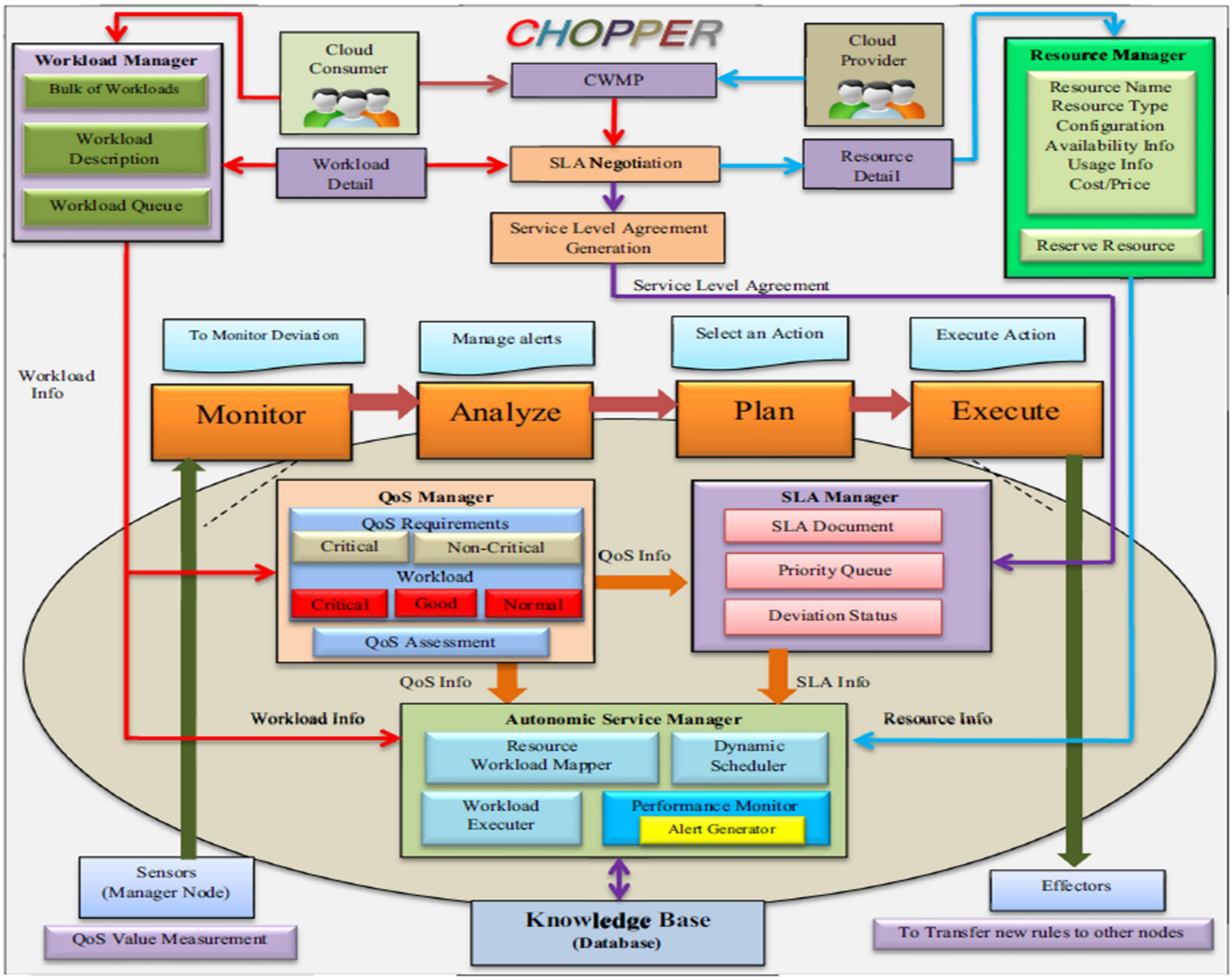

Figure. 6 Chopper architecture

3.1.1 Cloud Workload Management Portal (CWMP)

Cloud consumers run workloads on the CWMP via the web. Consumers authenticate and request resources and SLAs here. To achieve SLAs, CHOPPER optimizes resource distribution and corresponds with cloudy customer needs.

3.1.2. Workload Manager

Cloud deployment workload characteristics are assessed by this unit. It sorts workloads by QoS needs such security, network, and load fluctuation. Workload priority and QoS determine execution queue.

3.1.3. Resource Manager

The Resource Manager logs available and reserved resources. Based on SLA and other QoS factors, it optimizes resource use and meets workload demands. Dynamic alignment maximizes efficiency and minimizes waste.

3.1.4. Service Level Agreement (SLA) Management

Consumers and cloud providers agree on QoS through SLAs. CHOPPER dynamically negotiates these agreements to match resource capabilities with consumer expectations, enabling flexible and responsive resource management.

3.1.5. QoS Manager

To maintain and monitor QoS, this manager prioritizes jobs by urgency and resource needs. It adjusts resource allocations to dynamic cloud conditions and completes workloads on time and with QoS. Missed deadlines will be handled quickly using available resources. Additional executions or the reserved pool are used (Fig.6). Initial Key QoS determines every workload

submission. if the following matters. Good or typical work

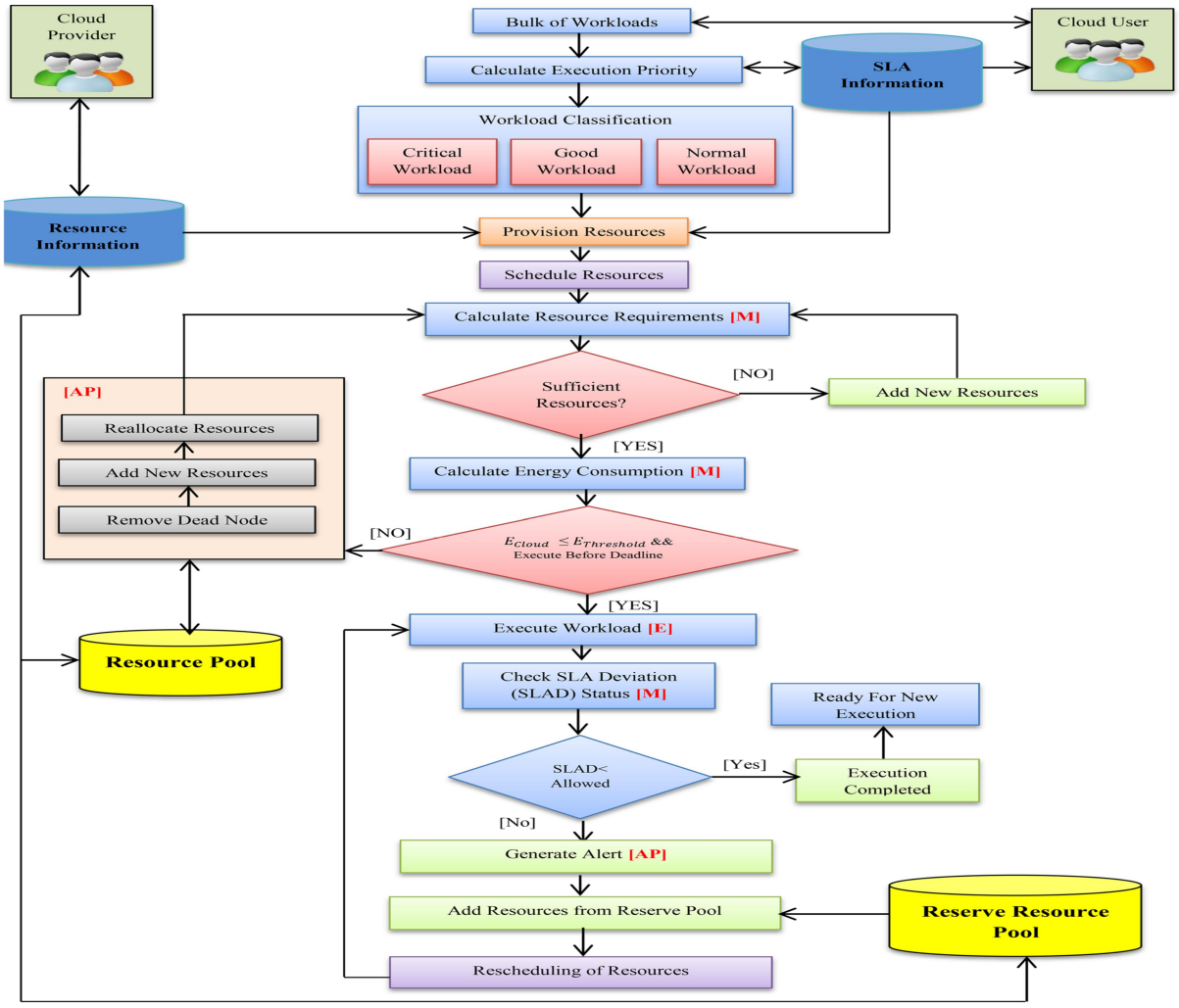

Figure 7: CHOPPER workload execution

3.1.6. Performance and Operational Efficiency

CHOPPER shown in figure 7 integrates several operational modules to ensure compliance with SLAs and optimal resource utilization:

Dynamic Scheduler and Workload Executor: These components work in tandem to schedule and execute workloads based on real-time resource availability and workload priority.

QoS Value Measurement and Adjustment: Continuous monitoring and adjustment mechanisms ensure that all operational parameters remain within defined thresholds.

Alert Management: A smart alert system detects SLA violations or resource shortages, triggering quick resource and task management modifications. And the below Figure. 8 shows Resource scheduling flowchart

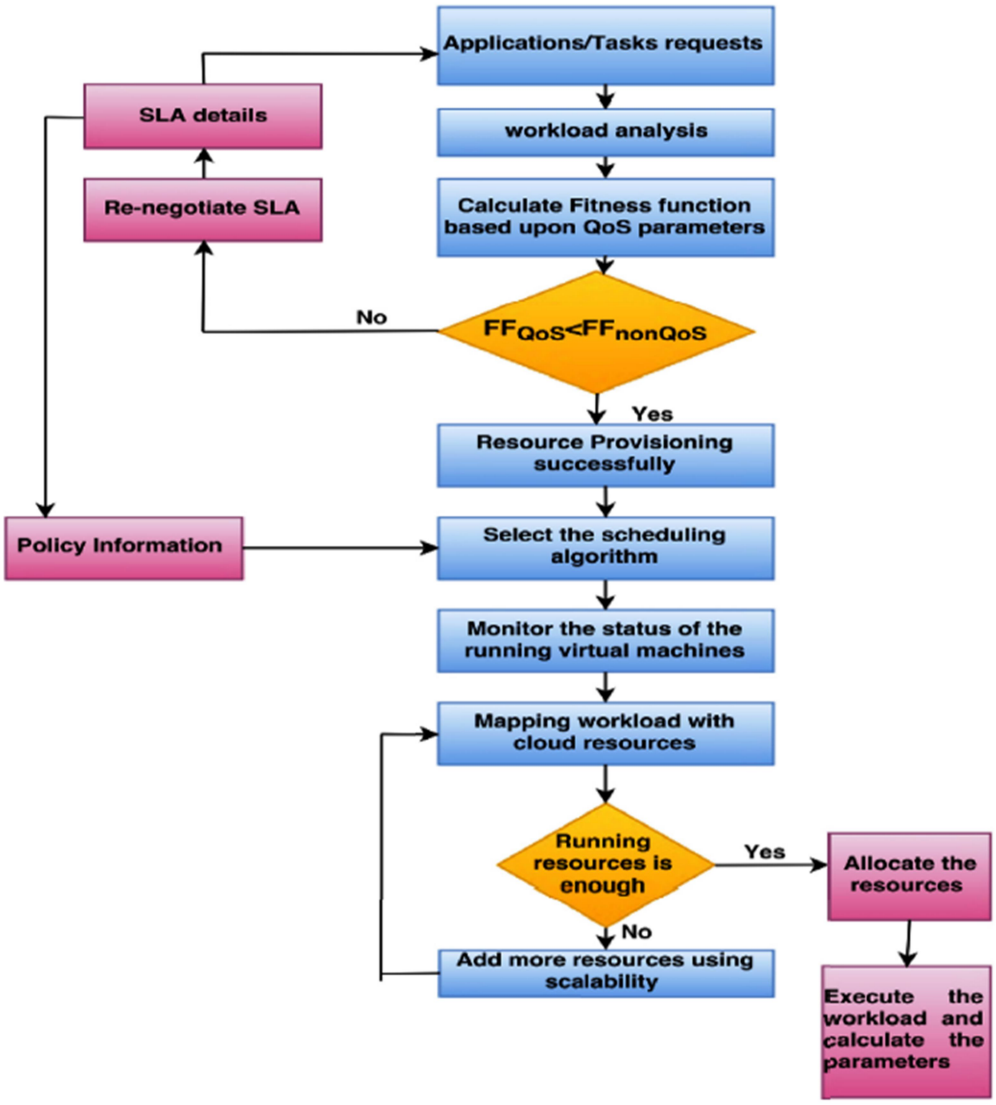

Figure. 8: Resource scheduling flowchart.

Hybrid Machine learning algorithms for Virtual Machine (VM)

Optimize cloud computing systems for VM Failure Detection, Dynamic Task Rescheduling, and QoS-Based Performance Management with hybrid machine learning. Improved energy consumption, execution times, and QoS boost cloud efficiency. This approach examines CPU, memory, network traffic, energy, and task execution cloud performance. Train hybrid supervised-unsupervised machine learning models with performance measures. Using performance data trends and anomalies, supervised models predict real-time VM failures. In failure, unsupervised learning models dynamically reschedule jobs to optimize resource distribution and reduce system downtime. VM faults are discovered and jobs rescheduled in real time via hybrid machine learning [49]. Advanced AI algorithms detect defects and rapidly adjust resource needs, maximizing resource consumption and reducing operational inefficiencies. Dependability, availability, security, and performance QoS are used. These indicators assist the system meet user expectations during workload swings or failures. AI-based optimization tools alter energy usage and job execution durations to improve energy efficiency and service performance. To guarantee

system functionality, experiments will measure VM failure detection accuracy, resource utilization, job execution time, and energy consumption. A comparison of system performance before and after hybrid machine learning. Increased energy economy, execution speed, and VM failure management should raise cloud service performance and management [50]. In [51] present a robust fault tolerant framework for the prediction of failures of virtual machines in advance and task scheduling. A segmented form regressive Q learning framework is designed and its main goal is to predict which virtual machine is likely to fail and cause service disruptions. The average failure rate of each of the virtual machines is calculated, considering the number of tasks submitted to virtual machines and the number of tasks successfully completed at the virtual machine. In [52] provide discussion on fault classification in the virtual machine architecture using deep learning. The conventional TabNet neural network architecture is enhanced to deal with the tabular form of input data. The decision tree algorithm is combined with the basic architecture of the neural network for fault detection. The virtual machine architecture is composed of three important components.

The Python code optimizes cloud computing systems for Virtual Machine (VM) failure detection and performance management using many supervised machine learning models. Predictive models are trained using cloud performance metrics including CPU, memory, and network traffic. Historical performance data is used to train Linear Regression, K-Nearest Neighbors (KNN), Support Vector Regressor (SVR), Boost, Light, and Random Forest models to detect VM failure abnormalities. The system may forecast VM failures in real time using these models to recognize VM behavior trends. Though it may struggle with non-linear data, Linear Regression is a straightforward, interpretable model for continuous predictions. Anomaly detection is simple with KNN, which predicts based on neighboring data points. SVR and boost manage more complex, non-linear interactions, which is important for detecting tiny VM performance changes that may indicate failure. The non-linear interactions and feature values of tree-based models like Light and Random Forest make them effective real-time failure detectors. However, the current failure detection method uses supervised learning, while dynamic task rescheduling uses unsupervised learning. To optimize resource use, clustering algorithms can dynamically disperse workloads and group tasks. This helps trained models adapt to changing workloads without labeled data, reducing downtime and improving cloud resource management. A hybrid system using machine learning methods anticipates VM faults and optimizes resource allocation through dynamic task rescheduling. Energy efficiency and QoS are supported by the code's hybrid machine learning system. The system forecasts resource utilization to optimize energy and task execution. Advanced AI-driven algorithms adapt the framework to changing resource demands, ensuring availability, reliability, and performance.

Energy Consumption Model: Cloud energy consumption is based on VM resource usage (CPU, memory, network bandwidth). Its representation:

$$E = \sum_{i=1}^{N} (\alpha \cdot \text{CPU}_i + \beta \cdot \text{Mem}_i + \gamma \cdot \text{Net}_i)$$

where:

- $\text{CPU}_i, \text{Mem}_i,$ and Net_i represent CPU usage, memory usage, and network traffic of the i^{th} VM.
- $\alpha, \beta,$ and γ are constants representing energy coefficients for CPU, memory, and network usage respectively.
- N is the total number of VMs in the system.

VM Failure Detection (Supervised Learning Model): To predict VM failure, the supervised machine learning model takes the performance metrics as inputs. The probability of a VM failure P_{fail} can be modeled as:

$$P_{\text{fail}}(\text{VM}_i) = f(\text{CPU}_i, \text{Mem}_i, \text{Net}_i, t_{\text{exec}_i})$$

where:

- f is the predictive function learned from historical data.
- t_{exec_i} represents the execution time of tasks on the i^{th} VM.
- $P_{\text{fail}}(\text{VM}_i)$ is the failure probability for the i^{th} VM.

Dynamic Task Rescheduling (Unsupervised Learning Model): Task rescheduling based on resource usage optimizes load balancing and reduces downtime. The rescheduling decision DDD can be represented as:

$$D = \arg\min_{R_j} \left(\sum_{k=1}^{M} w_k \cdot \mathrm{QoS}_k(R_j) \right)$$

where:

- R_j represents the set of resources available for rescheduling.
- $\mathrm{QoS}_k(R_j)$ are the Quality-of-Service metrics (e.g., reliability, availability) for resource R_j.
- w_k represents the weights assigned to different QoS metrics.
- M is the total number of QoS parameters being considered.

QoS-Based Optimization: The total Quality of Service $\mathrm{QoS}_{\mathrm{total}}$ for the system is defined as a weighted sum of individual QoS metrics, which can include reliability, availability, and security:

$$\mathrm{QoS}_{\mathrm{total}} = \sum_{k=1}^{M} w_k \cdot \mathrm{QoS}_k$$

where:

- w_k are the weights for each QoS metric.
- QoS_k are individual QoS values for different performance metrics.

Overall System Performance Improvement: The overall improvement I_{perf} in system performance after applying hybrid machine learning algorithms can be measured as:

$$I_{\mathrm{perf}} = \frac{P_{after} - P_{\mathrm{before}}}{P_{\mathrm{before}}} \times 10$$

where:

- P_{after} and P_{before} represent system performance metrics (energy consumption, task execution time, etc.) after and before applying the hybrid algorithms.

Energy Efficiency Optimization: The goal is to minimize energy consumption E while maintaining QoS standards. This optimization problem can be written as:

$$\min E \ \ subject\ to\ \ QoS_{totoal} \geq QoS_{min}$$

where:

- QoS_{min} is the minimum acceptable QoS level.

By incorporating these equations into the framework, the system can predict VM failures, optimize task rescheduling, and manage energy consumption effectively while maintaining high levels of QoS.

3.2 Research Method Protocol

The research protocol adopted in this study is illustrated through the following sequential steps to ensure the reproducibility of the proposed framework:

Step 1: Dataset Collection A cloud computing dataset containing CPU utilization, memory usage, network traffic, energy consumption, task execution time, task priority, and VM status was collected from a synthetic cloud environment.

Step 2: Data Preprocessing The collected dataset was cleaned by handling missing values, removing inconsistencies, and normalizing numerical attributes to the range of 0–1.

Step 3: Feature Engineering Relevant features associated with VM performance and resource utilization were selected, including CPU usage, memory usage, network traffic, and execution metrics.

Step 4: Model Development Six machine learning models, namely Linear Regression, K-Nearest Neighbors (KNN), Support Vector Regression (SVR), XGBoost, LightGBM, and

Random Forest, were developed and trained using historical cloud performance data.

Step 5: Hybrid Model Construction A hybrid framework combining LightGBM and Random Forest was developed to improve prediction accuracy and support real-time VM failure detection.

Step 6: VM Failure Detection The trained models continuously monitored VM performance metrics and predicted potential failures based on abnormal resource utilization patterns.

Step 7: Dynamic Task Rescheduling Upon detecting a potential VM failure, the proposed framework dynamically rescheduled tasks to alternative virtual machines to minimize downtime and maintain service availability.

Step 8: QoS Evaluation The framework evaluated Quality of Service parameters, including reliability, availability, energy consumption, execution time, and SLA compliance.

Step 9: Performance Assessment The prediction performance of all models was compared using Root Mean Square Error (RMSE), and the effectiveness of the proposed framework was analyzed based on resource utilization and QoS improvements.

4. RESULT AND DISCUSSION

This study analyzes cloud computing performance indicators to show how tasks and procedures affect system efficiency. This dataset shows system stress from CPU, memory, and network traffic. Instruction count and power usage show system energy and resource needs. Energy efficiency and execution time indicate system performance. Task type, priority, and status imply urgency. These features evaluate workload effects on system performance and resource allocation. Dataset-based machine learning models boost cloud execution speed and energy efficiency. Data originated from a synthetic cloud system that simulates several cloud computing scenarios. Scholars and developers can study how machine learning can improve cloud infrastructure performance and energy efficiency for cloud resource management. Cloud virtual machine management performance indicators included in the dataset. CPU, memory, network, and power utilization define system load and resources. Number of instructions, execution time, and energy efficiency measure cloud task computing performance and energy utilization. Normalized numeric properties are 0–1. In system performance evaluation, task type, priority, and status indicate task nature and urgency. Timestamp gaps indicate data gaps. Some VM-specific vibes contain missing values. This dataset examines resource utilization and VM failure prediction machine learning algorithms to optimize cloud computing. Cloud workload performance and energy efficiency are assessed using resource and task data. Missing data analysis must be done correctly for accuracy.

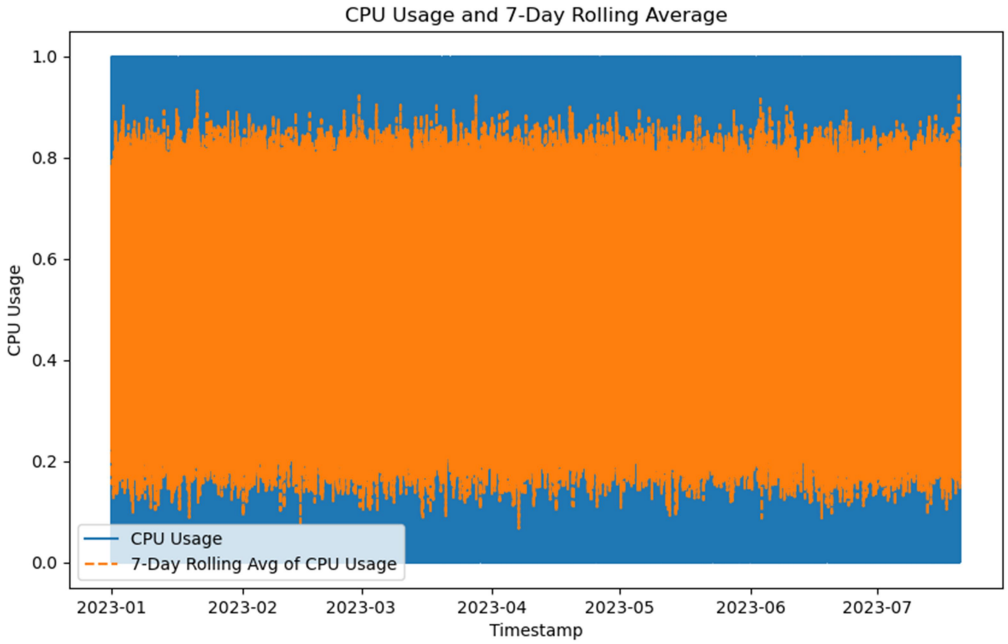

Figure 9: CPU Usage and 7-Day Rolling Average

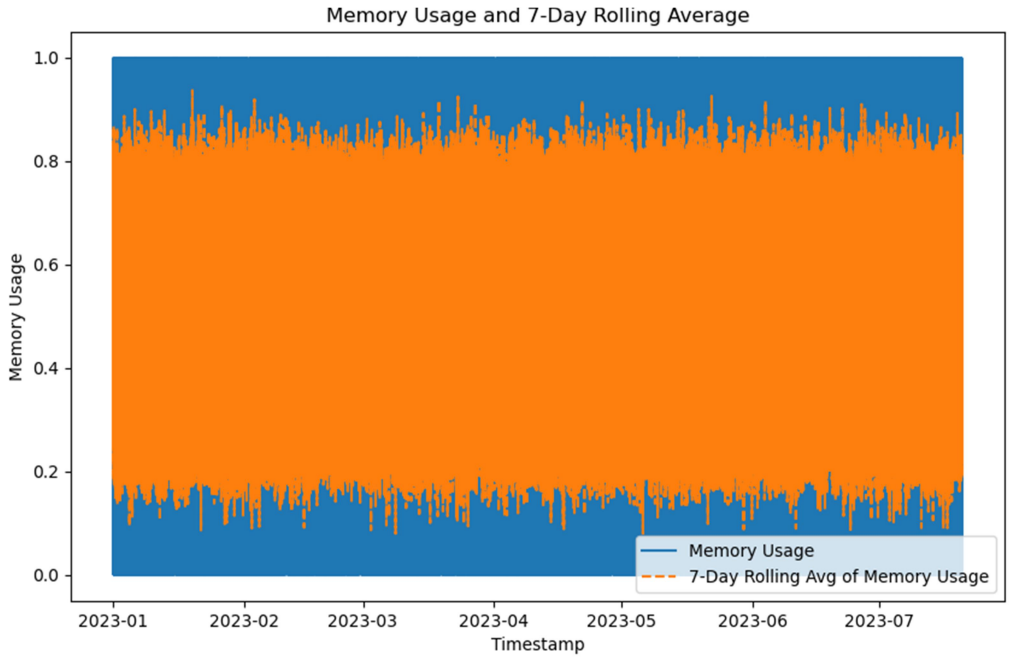

Figure 10: Memory Usage and 7-Day Rolling Average

Figure 9 shows CPU usage and 7-day rolling averages. Dashed line shows rolling average, which smooths short-term volatility to reflect long-term tendency. Primary line shows CPU use throughout time. While the rolling average shows CPU usage stability, this shows resource demand trends like high or low CPU utilization. Figure 10 shows memory utilization and 7-day rolling average. The solid line displays real-time memory consumption, whereas the dotted line shows rolling average without daily noise. Real usage shows cloud system resource constraints and memory demand trends compared to rolling average. These figures improve performance analysis, resource allocation, and system optimization.

Table 2: Root Mean Square Error (RMSE) comparison of machine learning models for CPU, memory, and network traffic prediction.

Method	CPU Usage RMSE	Memory Usage RMSE	Network Traffic RMSE
Linear Regression	0.2569	0.2527	0.2768
KNN	0.2984	0.3024	0.3007
SVR	0.259	0.2554	0.2779
XGBoost	0.2674	0.2593	0.2878
LightGBM	0.2659	0.2611	0.2888
RandomForest	0.2645	0.2583	0.2825
Combined (LightGBM + RandomForest)	0.2634	0.258	0.2842

Table 2 compares RMSE CPU, memory, and network traffic predictions from machine learning models. Linear Regression has a CPU consumption RMSE of 0.2569, one of the lowest error rates, however some models operate better. KNN exhibits higher RMSE at all measurements, including CPU consumption @ 0.2984 and memory usage @ 0.3024, indicating lesser accuracy than the other models. SVR surpasses KNN but trails leading models like XGBoost at 0.2590 CPU utilization RMSE. CPU usage RMSE values of 0.2674 and 0.2659 for XGBoost and LightGBM suggest balanced memory and network traffic estimations. RandomForest matches LightGBM in accuracy with 0.2645 CPU and 0.2583 memory RMSEs. The best model is the Combined (LightGBM + RandomForest) with CPU and memory RMSEs of 0.2634 and 0.2580. This combined strategy predicts better than the individual models, making it the most accurate technique.

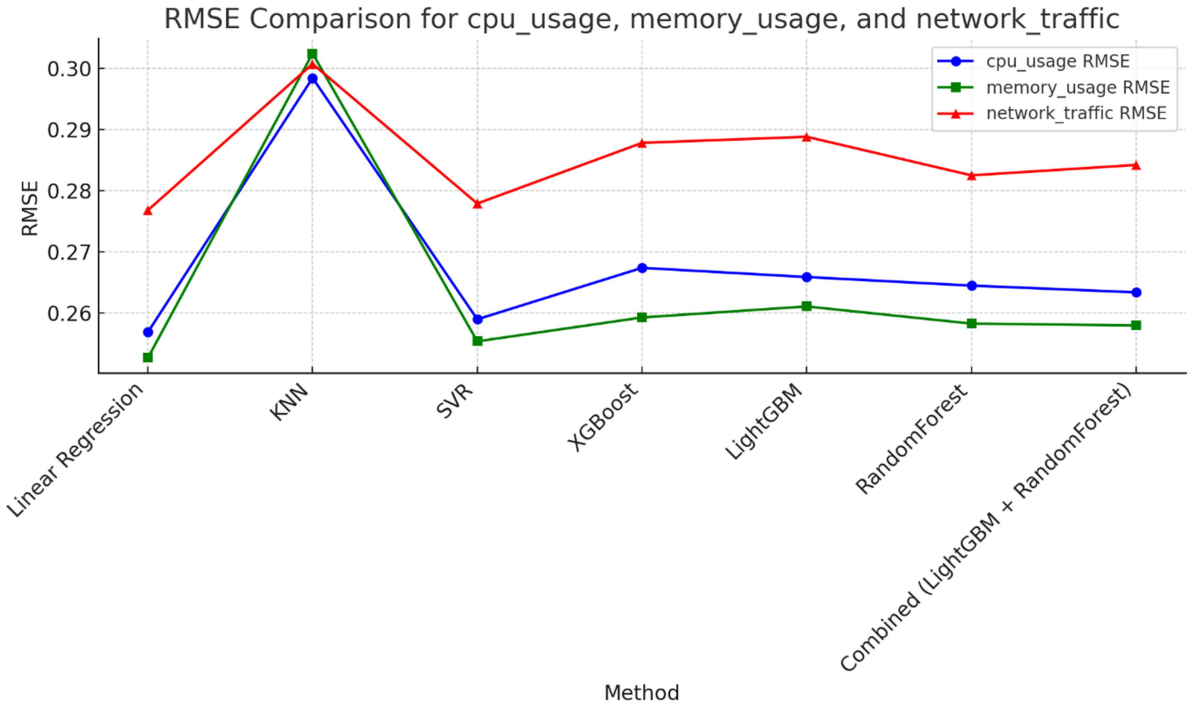

Figure 11: RMSE Comparison for cpu_usage, memory_usage, and network_traffic

Figure 11 demonstrates the RMSE for CPU, memory, and network traffic prediction using different machine learning methods. X-axis machine learning models include Linear Regression, KNN, SVR, boost, LightGBM, Random Forest, and a combination. Better model accuracy is represented by lower y-axis RMSE. This plot depicts CPU RMSE as blue circles, showing Linear Regression having the fewest errors and KNN the most. Green squares reflect memory utilization RMSE, with KNN worst and Light + RandomForest best. Red triangles show network traffic RMSE, like CPU and memory estimates. The combination model shines again. LightGBM + Random Forest predicts cloud performance better with lower RMSE across all

criteria, as seen in the graphic. We compared Linear Regression, KNN, SVR, XGBoost, LightGBM, RandomForest, and both across CPU, memory, and network traffic in cloud computing. We examined these models' Root Mean Square Error (RMSE) values for resource consumption prediction and cloud performance management accuracy. Table 2, Figures 9, 10, and 11 show these models' broad performance. Linear Regression exhibited low CPU and memory RMSE, although SVR, XGBoost, and RandomForest did better. A basic model like KNN has greater RMSE values across all measures, making it unsuitable for cloud resource prediction. SVR predicted well in all three areas. RMSE is superior for tree-based models like XGBoost and LightGBM. LightGBM and RandomForest ranked best with the lowest RMSE across all measures. Since both models improve forecast accuracy, this hybrid method is suitable for complex cloud performance management. Figures 9, 10, and 11 confirm. The 7-day rolling averages for CPU and memory utilization in Figures 9 and 10 show how the models smooth short-term swings to capture long-term patterns. Figure 11 compares model performance using indicator RMSE values.

4.1 Comparison with Existing Studies and Novel Contributions

Existing studies have primarily focused on individual aspects of cloud computing, including workload prediction, resource provisioning, energy optimization, and QoS-aware scheduling and it was given in table 3. For example, Yang and Chen [38] and Liu et al. [39] concentrated on resource usage prediction and container scheduling, whereas Tuli et al. [41] emphasized sustainable resource management and energy efficiency. Similarly, Bi et al. [44] and Xie et al. [45] focused mainly on workload and resource load prediction. However, these studies do not provide an integrated framework capable of simultaneously performing VM failure prediction, dynamic task rescheduling, and QoS-aware resource management. In contrast, the proposed study introduces a unified hybrid machine learning framework that combines supervised and unsupervised learning techniques to achieve real-time VM failure detection, adaptive task rescheduling, and intelligent QoS optimization. Furthermore, the proposed framework integrates multiple cloud performance metrics, including CPU utilization, memory usage, network traffic, and energy consumption, thereby providing a comprehensive and scalable solution for dynamic cloud environments.

Table 3. Comparative Analysis of Existing Studies and the Proposed Hybrid Machine Learning Framework

Study	**VM Failure Detection**	**Dynamic Task Rescheduling**	**QoS Management**	**Hybrid ML**	**Integrated Framework**
Yang and Chen [38]	✗	✗	Partial	✓	✗
Liu et al. [39]	✗	✗	Partial	✗	✗
Tuli et al. [41]	✗	✗	✓	✓	✗
Bi et al. [44]	✗	✗	✗	✓	✗
Xie et al. [45]	✗	✗	✗	✓	✗
Proposed Method	✓	✓	✓	✓	✓

5. CONCLUSION

A study showed that advanced machine learning models estimate cloud CPU, memory, and network traffic better. LightGBM and Random Forest had lowest RMSE. Cloud performance projections improved with both models. This paper suggests employing these models for real-time VM failure detection and resource allocation to optimize cloud operations. Hybrid model study may improve cloud performance prediction. Complex model combinations or unsupervised learning may enable dynamic task rescheduling. To boost machine learning, cloud performance indicators should incorporate storage and I/O. Applying these models to real-world cloud systems may improve forecasting.

5.1 Limitations and Open Research Issues

Despite the promising results obtained by the proposed hybrid machine learning framework,

several limitations remain. First, the experimental evaluation was conducted using a synthetic cloud dataset, which may not fully capture the complexity and heterogeneity of real-world cloud environments. Second, the proposed framework primarily focuses on CPU usage, memory utilization, and network traffic, while other important parameters such as storage I/O, latency, and security-related metrics were not considered. Third, the study evaluated the framework in a centralized cloud setting and did not investigate its applicability in multi-cloud, edge-cloud, or federated cloud environments. Furthermore, the computational overhead and scalability of the proposed hybrid model under extremely large-scale cloud infrastructures were not extensively analyzed. Future research may address these limitations by incorporating real-time cloud datasets, advanced deep learning and reinforcement learning techniques, additional QoS parameters, and deployment in large-scale heterogeneous cloud environments to further improve fault tolerance and intelligent resource management.

Scientific Contributions of the Study

- Development of a unified hybrid machine learning framework for VM failure detection, task rescheduling, and QoS management.
- Integration of supervised and unsupervised learning techniques for adaptive cloud resource management.
- Improvement in prediction accuracy through the hybrid LightGBM-Random Forest model.
- Enhancement of cloud reliability, energy efficiency, and SLA compliance.
- Provision of a scalable framework that addresses limitations of existing state-of-the-art solutions.

REFERENCES

[1]. Kumar M, Sharma SC. Deadline constrained based dynamic load balancing algorithm with elasticity in cloud environment. Computers & Electrical Engineering. 2018 Jul 1;69:395-411.

[2]. Madni SH, Abd Latiff MS, Coulibaly Y. Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities. Journal of Network and Computer Applications. 2016 Jun 1;68:173-200.

[3]. Hu H, Wen Y, Chua TS, Li X. Toward scalable systems for big data analytics: A technology tutorial. IEEE access. 2014 Jun 24;2:652-87.

[4]. Demchenko Y, De Laat C, Membrey P. Defining architecture components of the Big Data Ecosystem. In2014 International conference on collaboration technologies and systems (CTS) 2014 May 19 (pp. 104-112). IEEE.

[5]. Yi X, Liu F, Liu J, Jin H. Building a network highway for big data: architecture and challenges. Ieee Network. 2014 Jul 24;28(4):5-13.

[6]. Schroeder B, Gibson GA. The computer failure data repository (CFDR). InWorkshop on Reliability Analysis of System Failure Data (RAF'07), MSR Cambridge, UK 2007 Mar.

[7]. Alworafi MA, Dhari A, El-Booz SA, Nasr AA, Arpitha A, Mallappa S. An enhanced task scheduling in cloud computing based on hybrid approach. InData Analytics and Learning: Proceedings of DAL 2018 2019 (pp. 11-25). Springer Singapore.

[8]. Kanwal S, Iqbal Z, Al-Turjman F, Irtaza A, Khan MA. Multiphase fault tolerance genetic algorithm for vm and task scheduling in datacenter. Information Processing & Management. 2021 Sep 1;58(5):102676.

[9]. Ali A, Iqbal MM. A cost and energy efficient task scheduling technique to offload microservices based applications in mobile cloud computing. IEEE Access. 2022 Apr 28;10:46633-51.

[10]. Bharany S, Badotra S, Sharma S, Rani S, Alazab M, Jhaveri RH, Gadekallu TR. Energy efficient fault tolerance techniques in green cloud computing: A systematic survey and taxonomy. Sustainable Energy Technologies and Assessments. 2022 Oct 1;53:102613.

[11]. Kaur S, Bagga P, Hans R, Kaur H. Quality of service (QoS) aware workflow scheduling (WFS) in cloud computing: A systematic review. Arabian Journal for Science and Engineering. 2019 Apr 1;44:2867-97.

[12]. Manvi SS, Shyam GK. Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. Journal of network and computer applications. 2014 May 1;41:424-40.t

[13]. Jula A, Sundararajan E, Othman Z. Cloud computing service composition: A systematic literature review. Expert systems with applications. 2014 Jun 15;41(8):3809-24.

[14]. Zhang Q, Cheng L, Boutaba R. Cloud computing: state-of-the-art and research

challenges. Journal of internet services and applications. 2010 May;1:7-18. Dehghani M, Montazeri Z, Trojovská E, Trojovský P. Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. Knowledge-Based Systems. 2023 Jan 10;259:110011.

[15]. Zhong Z, Xu M, Rodriguez MA, Xu C, Buyya R. Machine learning-based orchestration of containers: A taxonomy and future directions. ACM Computing Surveys (CSUR). 2022 Sep 14;54(10s):1-35.

[16]. Hsieh SY, Liu CS, Buyya R, Zomaya AY. Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers. Journal of Parallel and Distributed Computing. 2020 May 1;139:99-109.

[17]. Jeong B, Baek S, Park S, Jeon J, Jeong YS. Stable and efficient resource management using deep neural network on cloud computing. Neurocomputing. 2023 Feb 7;521:99-112.

[18]. Mohammed B, Kiran M, Maiyama KM, Kamala MM, Awan IU. Failover strategy for fault tolerance in cloud computing environment. Software: Practice and Experience. 2017 Sep;47(9):1243-74.

[19]. Sanaj MS, Prathap PJ. An efficient approach to the map-reduce framework and genetic algorithm-based whale optimization algorithm for task scheduling in cloud computing environment. Materials Today: Proceedings. 2021 Jan 1;37:3199-208.

[20]. Kruekaew B, Kimpan W. Enhancing of artificial bee colony algorithm for virtual machine scheduling and load balancing problem in cloud computing. International Journal of Computational Intelligence Systems. 2020 Jan;13(1):496-510.

[21]. Neelima P, Reddy AR. An efficient load balancing system using adaptive dragonfly algorithm in cloud computing. Cluster Computing. 2020 Dec;23(4):2891-9.

[22]. Qin Y, Wang H, Yi S, Li X, Zhai L. An energy-aware scheduling algorithm for budget-constrained scientific workflows based on multi-objective reinforcement learning. The Journal of Supercomputing. 2020 Jan;76:455-80.

[23]. Jing W, Zhao C, Miao Q, Song H, Chen G. QoS-DPSO: QoS-aware task scheduling for cloud computing system. Journal of Network and Systems Management. 2021 Jan;29:1-29.

[24]. Amer DA, Attiya G, Zeidan I, Nasr AA. Elite learning Harris hawks optimizer for multi-objective task scheduling in cloud computing. The Journal of Supercomputing. 2022 Feb;78(2):2793-818.

[25]. Garg N, Singh D, Singh Goraya M. Deadline aware energy-efficient task scheduling model for a virtualized server. SN Computer Science. 2021 May;2:1-5.

[26]. Amiri M, Mohammad-Khanli L. Survey on prediction models of applications for resources provisioning in cloud. Journal of Network and Computer Applications. 2017 Mar 15;82:93-113.

[27]. Zhang J, Huang H, Wang X. Resource provision algorithms in cloud computing: A survey. Journal of network and computer applications. 2016 Apr 1;64:23-42.

[28]. Smanchat S, Viriyapant K. Taxonomies of workflow scheduling problem and techniques in the cloud. Future Generation Computer Systems. 2015 Nov 1;52:1-2.

[29]. Dave YP, Shelat AS, Patel DS, Jhaveri RH. Various job scheduling algorithms in cloud computing: A survey. InInternational Conference on Information Communication and Embedded Systems (ICICES2014) 2014 Feb 27 (pp. 1-5). IEEE.

[30]. Radha K, Rao B, Babu S, Rao K, Reddy V, Saikiran P. Allocation of resources and scheduling in cloud computing with cloud migration. International Journal of Applied Engineering Research. 2014;9(19):5827-37.

[31]. Nandhakumar C, Ranjithprabhu K. Heuristic and meta-heuristic workflow scheduling algorithms in multi-cloud environments—a survey. In2015 International Conference on Advanced Computing and Communication Systems 2015 Jan 5 (pp. 1-5). IEEE.

[32]. Kalra M, Singh S. A review of metaheuristic scheduling techniques in cloud computing. Egyptian informatics journal. 2015 Nov 1;16(3):275-95.

[33]. Masdari M, Salehi F, Jalali M, Bidaki M. A survey of PSO-based scheduling algorithms in cloud computing. Journal of Network and Systems Management. 2017 Jan;25(1):122-58.

[34]. Hall J, Andrews J. VMware certified professional data center virtualization on vSphere 6.7 study guide: exam 2V0-21.19. John Wiley & Sons; 2020 Aug 14.

[35]. Chauhan SS, Pilli ES, Joshi RC, Singh G, Govil MC. Brokering in interconnected cloud computing environments: A survey. Journal of Parallel and Distributed Computing. 2019 Nov 1;133:193-209.

[36]. Bittencourt LF, Goldman A, Madeira ER, da Fonseca NL, Sakellariou R. Scheduling in distributed systems: A cloud computing

perspective. Computer science review. 2018 Nov 1;30:31-54.

[37]. Tran CH, Bui TK, Pham TV. Virtual machine migration policy for multi-tier application in cloud computing based on Q-learning algorithm. Computing. 2022 Jun;104(6):1285-306.

[38]. Yang Y, Chen L. Design of kubernetes scheduling strategy based on LSTM and grey model. In2019 IEEE 14th International Conference on Intelligent Systems and Knowledge Engineering (ISKE) 2019 Nov 14 (pp. 701-707). IEEE.

[39]. Liu J, Wang S, Zhou A, Xu J, Yang F. SLA-driven container consolidation with usage prediction for green cloud computing. Frontiers of Computer Science. 2020 Feb;14:42-52.

[40]. Mehta HK, Harvey P, Rana O, Buyya R, Varghese B. WattsApp: Power-aware container scheduling. In2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC) 2020 Dec 7 (pp. 79-90). IEEE.

[41]. Tuli S, Gill SS, Xu M, Garraghan P, Bahsoon R, Dustdar S, Sakellariou R, Rana O, Buyya R, Casale G, Jennings NR. HUNTER: AI based holistic resource management for sustainable cloud computing. Journal of Systems and Software. 2022 Feb 1;184:111124.

[42]. Carvalho M, Macedo DF. Container scheduling in co-located environments using QoE awareness. IEEE Transactions on Network and Service Management. 2023 Feb 10;20(3):3247-60.

[43]. Calheiros RN, Masoumi E, Ranjan R, Buyya R. Workload prediction using ARIMA model and its impact on cloud applications' QoS. IEEE transactions on cloud computing. 2014 Aug 21;3(4):449-58.

[44]. Bi J, Ma H, Yuan H, Zhang J. Accurate prediction of workloads and resources with multi-head attention and hybrid LSTM for cloud data centers. IEEE Transactions on Sustainable Computing. 2023 Mar 20;8(3):375-84.

[45]. Xie Y, Jin M, Zou Z, Xu G, Feng D, Liu W, Long D. Real-time prediction of docker container resource load based on a hybrid model of ARIMA and triple exponential smoothing. IEEE Transactions on Cloud Computing. 2020 Apr 22;10(2):1386-401.

[46]. Arul Xavier VM, Annadurai S. Chaotic social spider algorithm for load balance aware task scheduling in cloud computing. Cluster Computing. 2019 Jan 16;22(Suppl 1):287-97.

[47]. Alharkan I, Saleh M, Ghaleb MA, Kaid H, Farhan A, Almarfadi A. Tabu search and particle swarm optimization algorithms for two identical parallel machines scheduling problem with a single server. Journal of King Saud University-Engineering Sciences. 2020 Jul 1;32(5):330-8.

[48]. Sollfrank M, Loch F, Denteneer S, Vogel-Heuser B. Evaluating docker for lightweight virtualization of distributed and time-sensitive applications in industrial automation. IEEE Transactions on Industrial Informatics. 2020 Sep 8;17(5):3566-76.

[49]. Butt UA, Mehmood M, Shah SB, Amin R, Shaukat MW, Raza SM, Suh DY, Piran MJ. A review of machine learning algorithms for cloud computing security. Electronics. 2020 Aug 26;9(9):1379.

[50]. Sun T, Xiao C, Xu X. A scheduling algorithm using sub-deadline for workflow applications under budget and deadline constrained. Cluster Computing. 2019 May;22(Suppl 3):5987-96.

[51]. Rani, S.S.; Alfawaz, O.; Khedr, A.M. A robust fault-tolerant framework for VM failure predication and efficient task scheduling in dynamic cloud environments. *J. Netw. Comput. Appl.* **2025**, *244*, 104340.

[52]. Rawat, A.; Mishra, A.; Alsharif, H.; Alharthi, R.M.; Gupta, B.B. Fault classification in the architecture of virtual machine using deep learning. *Sci. Rep.* **2025**, *15*, 34469.