Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 16s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

An Efficient Dynamic Load Balancing and Resource Provisioning Scheme for Cloud Computing Environments

Bhargavi Ranga[*1], Murali Mohan V[2]

[1,2]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India.
[1]Email ID: 2301050201@kluniversity.in, [2] Email ID: muralimohan310@kluniversity.in

Cite this paper as: Bhargavi Ranga, Murali Mohan V, (2025) An Efficient Dynamic Load Balancing and Resource Provisioning Scheme for Cloud Computing Environments. *Journal of Neonatal Surgery*, 14 (16s), 134-143.

ABSTRACT
Effective resource management is essential for maximizing performance and guaranteeing adherence to service level agreements (SLAs) in the ever-changing world of cloud computing. While reactive fault tolerance techniques usually handle problems only after they arise, resulting in downtime and inefficiencies, traditional load balancing techniques frequently find it difficult to adjust to changing demands. This study suggests a novel hybrid approach that uses machine learning techniques to combine proactive fault tolerance mechanisms with dynamic load balancing. The system predicts changes in workload and possible errors by examining real-time metrics and previous data, which allows for more efficient resource allocation and a reduction in SLA breaches. According to preliminary findings, resource utilization has improved by more than 80%, and fault recovery times have significantly decreased . By providing a thorough foundation for further study and opening the door for more robust cloud computing systems that put efficiency and dependability first, this work fills in the gaps in literature.

Keywords: *Cloud Computing, Resource Provisioning , Service Level Agreements , Machine Learning, Neural Network, Fault Tolerance.*

1. INTRODUCTION

Effective use of cloud resources, including storage, networking, and processing power, depends on reliable load-balancing systems. In order to solve possible issues like SLA (Service Level Agreement) violations, performance degradation, and resource underutilization, load balancing makes sure that workloads are distributed fairly among the resources that are available. Ineffective load distribution can compromise the overall Quality of Service (QoS) by causing server overloading, execution errors, and resource waste.

Using sophisticated load-balancing techniques is essential to overcoming these obstacles. By allocating resources optimally, these tactics guarantee higher server utilization and better QoS. The dynamic management of virtual machines (VMs), including their migration and distribution, is a popular strategy in this field for adaptive load balancing. This study expands on existing approaches by putting forward novel strategies to deal with current cloud environment issues.

1.0 Cloud Computing

The on-demand access to shared computer resources provided by cloud computing has completely changed how individuals and organization's handle, store, and analyze data. It enables users to grow resources dynamically while reducing expenses by doing away with the requirement for significant on-premises infrastructure. But for cloud computing to be effective, its three main components computation, networking, and storage—must be managed effectively to satisfy workloads' changing demands. In order to avoid bottlenecks, underutilization, and SLA breaches, load balancing a system that divides workloads evenly among resources is a crucial component of this management.

The term "cloud computing" describes the provision of computer services, such as networking, processing, and storage, via the internet. It gives consumers pay-as-you-go access to resources, providing previously unheard-of flexibility and scalability. Three main models are commonly used to provide cloud services:

Infrastructure as a Service (IaaS): Provides storage and servers that are virtualized (e.g., AWS EC2, Google Compute Engine).

Platform as a Service (PaaS): Offers an environment for developing and launching applications (e.g., Microsoft Azure App Services, Google App Engine).

Software as a Service (SaaS): Provides software programs (like Gmail and Salesforce) via the internet.

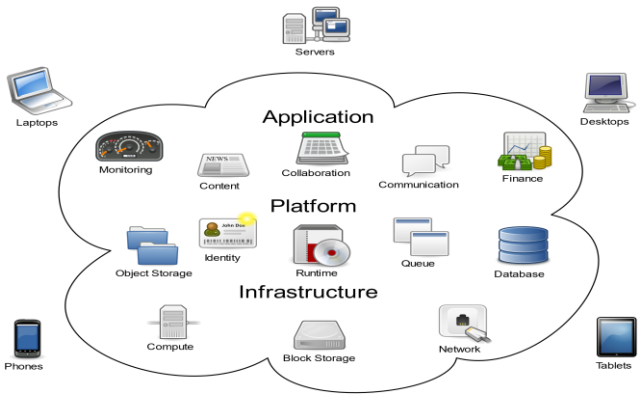

Figure 1 Cloud Computing Service Models

1.1 Importance of Load Balancing in Cloud Computing

In order to maximise cloud infrastructure performance, load balancing is essential. Uneven resource distribution can result in server overload, higher latency, and resource waste if load balancing is not done correctly.

Good load-balancing techniques guarantee:

- **Better QoS (Quality of Service):** Load balancing guarantees high service availability and reduces delays.
- **Optimal Resource Utilization:** The distribution of tasks minimises idle resources while preventing overload.
- **SLA Compliance:** It avoids outages that can result in fines under the terms of the contract. Dynamic task allocation via virtual machine (VM) distribution and migration is a common topic of research in this field. Modern methods seek to address issues like energy efficiency and real-time adaptation by utilising cutting-edge strategies like AI-based optimisations and predictive algorithms.

2. LITERATURE SURVEY

2.1 Dalia Abdulkareem Shafiq , Noor Zaman Jhanjhi , Azween Abdullah , And Mohammed A. Alzain [1]

Job scheduling is a key component of efficient load balancing in cloud settings, as the literature has shown. Task scheduling optimisation guarantees better system performance and scalability in addition to improving resource utilisation. An enhanced load balancing algorithm (LBA) is presented in this work in order to overcome the drawbacks of the current dynamic techniques. When compared to conventional Dynamic LBAs, the suggested algorithm performs better, reducing Make span significantly and improving resource utilisation by about 78%. Moreover, it exhibits strong flexibility in dynamic cloud settings with regularly varying user request durations and unpredictable job arrival patterns.In contrast to earlier approaches, the suggested technique efficiently handles heavier workloads while guaranteeing resource allocation. In order to reduce SLA breaches brought on by Virtual Machine (VM) limitations and guarantee the smooth completion of activities even in the face of high demand, it integrates dynamic resource reallocation techniques. The algorithm is now positioned as a scalable, dependable, and effective solution for contemporary, dynamic cloud computing environments thanks to these improvements.

2.2 Tawfeeg Mohmmed Tawfeeg , Adil Yousif , Alzubair Hassan , Samar M. Alqhtani , Rafik Hamza , Mohammed Bakri Bashir, And Awad Ali [2] As demonstrated by this thorough literature analysis, reactive fault tolerance mechanisms and dynamic load balancing strategies must be integrated to provide high availability, reliability, and effective resource utilisation in cloud computing settings. The results highlight the need for ongoing research and innovation to solve enduring difficulties, even though there has been substantial success in developing current frameworks and approaches. Frameworks for reactive fault tolerance need to change in order to properly manage a wider range of failure scenarios. By utilising cutting-edge methods like machine learning (ML) and deep learning (DL), predictive failure detection, automated recovery processes, and adaptive fault-handling approaches can be made possible. By enabling cloud systems to move from reactive to more proactive fault management, these features would improve resilience in dynamic contexts.To guarantee peak performance, dynamic load-balancing strategies must simultaneously concentrate on correcting imbalances among dispersed nodes. Advanced techniques that make use of deep learning algorithms can enhance resource scalability, optimise task scheduling, and dynamically forecast workload distributions.

2.3 Mayank Sohani and Shobhit Jain [3] Achieving effective load balancing among virtual machines (VMs) in heterogeneous cloud environments where a variety of computational resources and workloads must be managed concurrently is the main difficulty this work attempts to address. In order to address these issues, the PMHEFT algorithm presents a predictive model that predicts the future resource requirements of an application, allowing the cloud infrastructure to distribute resources in advance and dynamically. By distributing resources in line with expected workload demands, this proactive resource provisioning technique guarantees that the system will continue to be flexible, scalable, and effective.

2.4 Abdul Waheed, Munam Ali Shah, Syed Muhammad Mohsin [4] Task offloading, a crucial procedure in automotive networks, particularly for nodes with constrained processing capabilities, is the main topic of the essay. In the context of intelligent vehicle systems (IVS), where effective resource management is crucial for guaranteeing seamless operations, especially for delay-sensitive applications, it emphasises the need of offloading activities. The development of several computing paradigms, including mobile cloud computing (MCC) and multi-access edge computing (MEC), to overcome the computational difficulties faced by intelligent cars is covered in the paper. Real-time processing and the needs of contemporary, intelligent vehicle systems are supported by the offloading of computational chores to external resources made possible by these developments in computing and communication technology.

2.5 Bo Wang, Min Xie, Yingya Song, and Yu Cao [5] The study offers a thorough taxonomy for classifying work offloading in edge-cloud computing according to important criteria like goals, mobility, offloading tactics, and task categories. The review, which examines 71 relevant publications, identifies a number of issues and makes several suggestions for further study with the goal of maximising the use of edge-cloud computing in service delivery. The integration of decentralised offloading procedures, scheduling, task distribution, and offloading decisions are some of these difficulties. The report also identifies problems in assessing resource needs, controlling expenses, and successfully managing resource provisioning delays.The assessment also discusses the necessity of strong security measures, the possibilities of multi-cloud settings, and the challenges presented by user device mobility. It also highlights how important it is to enhance data monitoring over large edge-cloud networks and edge server infrastructures. According to the authors, their research will be especially helpful to academics and business experts who are interested in developing edge-cloud computing because it offers practical advice on how to get over current obstacles and promote innovation in the space.

2.6 Xiangbin Wen, Yuan Zheng [6] In order to improve resource management and overall performance, the study investigates the incorporation of artificial intelligence (AI) technologies into cloud computing settings. It provides a thorough analysis of several AI approaches, such as performance assessment tools and dependability models, and how they might be used to optimise cloud resources. The study shows how AI-driven methods may greatly increase the effectiveness and robustness of cloud services by tackling problems like workload management, system dependability, and resource allocation through in-depth experimental analysis and validation procedures. The study highlights AI's potential to transform cloud computing technologies by utilising it, especially in terms of optimising resource provisioning and service delivery. The results offer insightful information about how AI might be applied to improve cloud infrastructure scalability, anticipate system faults, and automate decision-making processes. This work offers a roadmap for future developments in cloud performance optimisation and service reliability, adding to the continuing conversation about how AI is influencing next-generation cloud services.

2.7 Michele De Donno, Koen Tange, And Nicola Dragoni [7] According to the authors, the cutting edge of contemporary computing paradigms is represented by the Internet of Things (IoT), cloud computing, edge computing, and fog computing. They highlight how hard it is for novices to understand how different paradigms differ, overlap, and depend on one another. This paper provides a thorough account of the development of Edge and Fog computing as well as current research trends, making it an essential starting point for scholars venturing into these fields. The authors carefully outline each paradigm's distinguishing traits, designs, and features, highlighting how they interact and how crucial fog computing is as the link between IoT, cloud, and edge computing. The study provides a thorough overview of outstanding problems and prospective topics for additional research by identifying open challenges and future research paths. The authors present this work as a vital resource for novices in the field, enabling deeper engagement and study into these quickly changing technologies, acknowledging the lack of such a consolidated resource in the body of current literature at the time of their research.

2.8 A. U. Rehman, Rui L. Aguiar, And João Paulo Barraca [8] The need for specialised knowledge in the design, development, and migration of cloud-based applications is anticipated to increase dramatically as the worldwide cloud computing business, which was estimated to be worth $250 billion in 2019, keeps expanding. This need will be further fuelled by projections that show 80% of organisations would have moved to cloud environments by 2025. Even with Wang et al. and Ding et al.'s encouraging task completion rates, a tiny portion of jobs still fail, highlighting the need for more reliable and effective fault-tolerant models to improve cloud infrastructure dependability. This survey offers a comprehensive analysis of cloud computing, covering its fundamental traits, advantages, deployment strategies, and architectural frameworks. It explores the crucial area of cloud fault tolerance by methodically classifying different kinds of faults, talking about mitigation strategies, performance indicators, and the frameworks and technologies that are currently in use. Additionally, it examines proactive and reactive fault management techniques, assessing how well each contributes to reducing interruptions and preserving service continuity.

2.9 Hana Eljak , Ashraf Osman Ibrahim , Fakhreldin Saeed , Ibrahim Abaker Targio Hashem , Abdelzahir Abdelmaboud , Hassan Jamil Syed , Anas Waleed Abulfaraj , Mohd Arfian Bin Ismail , And Abubakar Elsafi [9] The study's authors emphasise how technology is revolutionising online education, pointing out in particular how cloud computing is emerging as a game-changing paradigm. More flexible and scalable educational models are made possible by cloud computing, which supports distributed applications across multiple regions and provides on-demand, metered access to computing resources. The authors explore the use of cloud computing into e-learning systems in a thorough review of 154 academic publications from 2010 to 2022, highlighting the possibilities for remote learning and virtual work settings. Although there are many reviews available, the authors point out that there is a notable lack of practical applications despite the growing body of literature. This suggests that a more comprehensive integration of hardware, software, and security components is clearly necessary to guarantee the successful deployment of cloud-based learning systems. The report admits that although public cloud computing is an affordable option, data security is still a big worry, especially when handling sensitive data like student grades.

2.10 Ivana Zinno, Stefano Elefante, Lorenzo Mossucca, Claudio De Luca, Michele Manunta, Olivier Terzo, Riccardo Lanari, and Francesco Casu [10] The authors also note the challenges associated with processing vast amounts of Synthetic Aperture Radar (SAR) data, exemplified by the COSMO-SkyMed constellation and SENTINEL-1 satellites. They acknowledge that such data sets can present significant processing difficulties. One key issue arises from the use of parallel programs that simultaneously read and write from a shared storage volume, often resulting in a bottleneck when utilizing a common NFS-based storage architecture. To address this, the authors propose a distributed file system that minimizes network workload and reduces I/O bottlenecks by allocating data storage locally, improving overall performance. When applied to the designated number of nodes, the P-SBAS algorithm exhibits scalable performance in a cloud computing (CC) environment, according to the authors' conclusion. The cloud-based approach is especially useful for processing image time series from popular long-term C-band SAR sensors, such ENVISAT, they point out. For about USD 200, the cloud system can analyse a typical ENVISAT SAR picture time series in about 7 hours, making it possible to handle vast chunks of the ENVISAT archive efficiently in a condensed amount of time. Using a suitably large number of nodes accessible in cloud computing settings to process several ENVISAT SAR datasets at once significantly improves this efficiency.

3. METHODOLOGY

A hybrid dynamic load balancing technique and a fault tolerance mechanism driven by machine learning are introduced in the suggested study. By proactively controlling both load distribution and problem prediction, the objective is to improve fault tolerance recovery, decrease SLA (Service Level Agreement) breaches, and maximize resource utilization in cloud computing settings.

3.1 Research Gaps and Objectives

Load balancing and fault tolerance are frequently handled as distinct tasks in modern cloud computing settings, which results in inefficiencies, service outages, and resource waste. The approach seeks to close these disparities by:

- Reactive fault tolerance and dynamic load balancing are integrated to guarantee high availability.
- Machine learning is being used to forecast possible defects and resource loads.
- increasing adherence to performance, availability, and reliability criteria by broadening the scope of SLA parameters.

3.2 Data Collection and Preparation

3.2.1 Data Sources: Real-world data is available from sources including AWS Open Datasets, Azure Open Datasets, and Google Cloud Public Datasets. Logs on virtual machine usage, fault occurrences, resource utilisation, SLA adherence, and user request patterns are among these datasets.

- **Public Cloud Datasets:** Synthetic datasets can be created when there is a lack of enough public data. It is possible to customise data for particular testing requirements by using synthetic data, which is made to mimic load changes, task sizes, and problem types seen in real-world situations.
- **Synthetic Data:** Synthetic datasets can be created when there is a lack of enough public data. It is possible to customise data for particular testing requirements by using synthetic data, which is made to mimic load changes, task sizes, and problem types seen in real-world situations.

3.2.2 Data Features

- **Resource Utilization Metrics:** Disc I/O, network I/O, and CPU and memory utilisation are monitored at different times. These indicators highlight under- or over-utilized resources, which helps with load balancing decisions.
- **Task Execution Times:** calculates the time needed to complete tasks on different virtual machines. Under optimized resource allocation may be indicated by longer execution durations.

- **Historical Fault Tolerance:** Includes past malfunctions as well as related circumstances like excessive CPU use or network congestion. For fault prediction modelling, this information is crucial.
- **SLA Parameters:** comprises data such as downtime, throughput, and reaction time. SLAs are especially crucial for assessing how well the system fulfils user expectations.

3.2.3 Data Preprocessing:

- **Data Cleaning:** Inconsistencies, missing values, and anomalies in the data are addressed to guarantee precise machine learning training.
- **Feature Engineering:** Raw data is used to create new features. For instance:
  - **Load Variability:** Tracks changes in the demand for resources over time.
  - **Fault Patterns:** Identifies factors that frequently precede failures by classifying fault events according to resource measurements.
  - **SLA Violation Flags:** Response times and throughput statistics are used to create binary indicators of SLA adherence or violation.

Category	Size
vm_id	1799362
time_stamp	1799334
cpu_usage	1800962
memory_usage	1799490
network_traffic	1800519
power_consumption	1799729
num_executed_instructions	1800314
execution_time	1800173
energy_efficiency	1799958
task_type	1800038
task_priority	1800567
task_status	1799694
Total	**21600140**

Table 1 Cloud Performance Dataset

3.3 Machine Learning Models for Predictive Analytics

In this hybrid method, proactive load balancing and fault tolerance are based on predictive analytics.

3.3.1 Libraries:

- **Scikit-learn:** Perfect for interpretable and computationally efficient models for defect and load prediction, such as Random Forests and Decision Trees.
- **TenserFlow/PyTorch:** Convolutional neural networks (CNN) and recurrent neural networks (RNN), two deep learning models that are appropriate for complicated pattern identification in huge datasets, are made possible by these frameworks.

3.3.2 Model Development:

- **Load Prediction Model:** Uses previous data trends to forecast future resource loads. Regression models (such as Decision Trees and Linear Regression) or neural networks (such as RNNs) are used to accomplish this.
- **Fault Prediction Model:** Uses classification models (such as Random Forest and Support Vector Machine) trained on previous fault data to identify conditions that could result in faults. CNNs can be used to capture complex correlations between data features for enhanced detection.

3.3.3 Parameter Tuning:

- **Hyperparameter Tunning:** Model parameters are optimised using methods like Grid Search and Random Search, which strike a balance between computing efficiency and accuracy.
- **Evaluation Metrics:**
  - Accuracy for classification models (fault prediction).
  - For load prediction models, use Mean Absolute Error (MAE) or Mean Squared Error (MSE).

3.4 Hybrid Dynamic Load Balancing Algorithm

The goal of the load-balancing algorithm suggested in this study is to guarantee flawless performance and maximise resource utilisation. An explanation of its main phases is provided below for more clarity:

Monitoring of Resources

Input: Real-time resource usage measurements, such as network bandwidth, memory usage, and CPU utilization.
Process:

- To collect performance statistics, continuously monitor every node that is operational within the cloud infrastructure.
- Depending on the design, a distributed or centralized monitoring system may be employed.

Output: A profile of each node's dynamic resource usage.

Assessment of Load

Input: Information obtained from the resource monitoring phase.

Process: Use predetermined thresholds to assess each node's load. Nodes are categorised as balanced, overloaded, or underloaded based on SLA parameters or resource utilisation percentages.

Output: Categorisation list that groups all nodes according to the load they are now carrying.

Assignment and Scheduling of Tasks

Input: The list of load classifications and incoming task requests.

Process:

- Assign incoming tasks to nodes that are balanced or underloaded.
- To distribute resources as efficiently as possible, give priority to nodes that are not being used.
- When overload occurs, move virtual machines to less-loaded nodes or redistribute duties.
- When scheduling, take resource requirements, dependencies, and job priority into account.

Output: Guarantees the best possible load distribution is a task-node mapping.

Adaptive Dynamic Loading

Input: Current load conditions and performance indicators.

Process:

- Keep an eye out for imbalances in the system.
- When necessary, start resource scaling (up or down) or virtual machine migrations.
- Use predictive modelling to foresee future load trends and make proactive resource adjustments.

Output: A real-time, load-balanced cloud architecture with low latency and SLA compliance.

Feedback and Performance Assessment

Input: Performance indicators of the system after load balancing.

Process:

- Evaluate attained performance indicators in relation to QoS standards.
- Keep track of comments so the method can be improved iteratively.

Output: Knowledge for improving the algorithm to accommodate changing workload parameters.

Algorithm Code Snippet: A sample of Python code illustrating a simple implementation of the load balancing technique

can be seen below:

```
1. import numpy as np
2. import random
3. class VM:
    a. def __init__(self, id):
        i. self.id = id
        ii. self.current_load = 0
    b. def assign_task(self, task):
        i. self.current_load += task.load
    c. def load_balancing_algorithm(vms, tasks):
        i. for task in tasks:
            1. best_vm = min(vms, key=lambda vm: vm.current_load)
            2. best_vm.assign_task(task)
            3. return vms
4. # Simulating the load balancing process
5. vms = [VM(i) for i in range(5)]
6. tasks = [Task(random.randint(1, 10)) for _ in range(20)]
7. load_balancing_algorithm(vms, tasks)
```

- **Optimization Techniques:** The algorithm dynamically modifies load allocation to satisfy real-time demand and reduce SLA breaches by continually monitoring changes in virtual machine load.

3.5 Reactive Fault Tolerance Mechanisms

The following strategies are used by fault tolerance mechanisms to lessen the impact of failures:

- **Checkpointing:** Saves resources and minimises downtime by periodically recording a task's state and enabling it to restart from the most recent checkpoint in the event of failure.
- **Task Migration:** To avoid service interruption, jobs are transferred to other virtual machines when errors are detected. To prevent resource overloading on any one virtual machine, this technique works in tandem with load balancing.
- **Comparison with Baseline Models:** To verify gains in resource usage, fault recovery time, and SLA compliance, baseline comparisons are made (e.g., classical load balancing without fault tolerance).

3.6 Implementation Environment and Testing and Evaluation

- Cloud systems (AWS, Google Cloud, and Azure) are used for testing in order to replicate large-scale scenarios.
- Pandas and NumPy take care of data pretreatment and analysis, while libraries like Scikit-learn and TensorFlow make it easier to construct models.
- To ensure computational power for testing and fine-tuning the hybrid system, experiments are conducted on virtual machines with multi-core processors and at least 8GB of RAM.
- **Simulated Load and Fault Conditions:** To verify the model's scalability and resilience, performance is assessed in a variety of scenarios.
- **Evaluation Metrics:**
    - **Resource Utilization:** Aims for > 80% efficiency when measuring virtual machine use.
    - **Execution Time and Makespan:** Shorter makespan and shorter execution times for individual tasks.
    - **SLA Compliance:** Monitors SLA adherence and aims to cut infractions by 25%.
    - Fault Recovery Time: evaluates the system's fault recovery speed with the goal of increasing it by 30%.
- **Limitations and Assumptions**

- presupposes constant data availability for fault and load prediction.
- To properly handle new problem scenarios, more data would be needed as the system is optimized for common fault kinds.

4. EXPERIMENTAL RESULTS

CPU and Memory Usage Distribution

The histogram and density plot for CPU and memory utilisation are displayed in the CPU and Memory utilisation Distribution graph. This aids in the analysis of the dataset's resource usage trends.

The Distribution Graphs' Insights:

- **Skewness**: Whether resource utilisation is skewed towards higher or lower values is indicated by the distribution. While some jobs require substantially more resources than others, a positively skewed distribution indicates that most tasks use very little.
- **Peak Usage:** Finding average utilisation trends is made easier by the peaks, which show the most common amounts of resource usage.

Overlapping Patterns: Overlaps between CPU and memory density graphs can show workloads that have a proportional connection in resource demands.

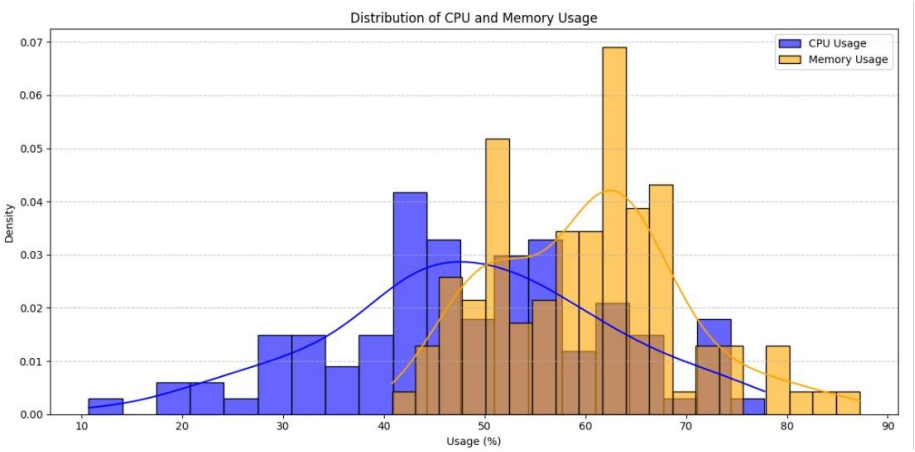

Figure 2 Distribution of CPU and Memory Usage

7-Day Average Rolling (Time Series Analysis)

Smoothed patterns in CPU and memory consumption over time are displayed on the 7-day rolling average graph. This method eliminates noise and draws attention to important patterns.

Insights from Rolling Average Graphs

- **Periodic Trends:** Predictable cycles, like increased resource consumption during peak operating hours or days, may be reflected in weekly trends.
- **Anomalies:** Significant departures from the norm may be a sign of anomalous activity or increases in resource usage.
- **Load Balancing Efficiency**: Finding recurring trends helps confirm that load-balancing algorithms are working as intended.

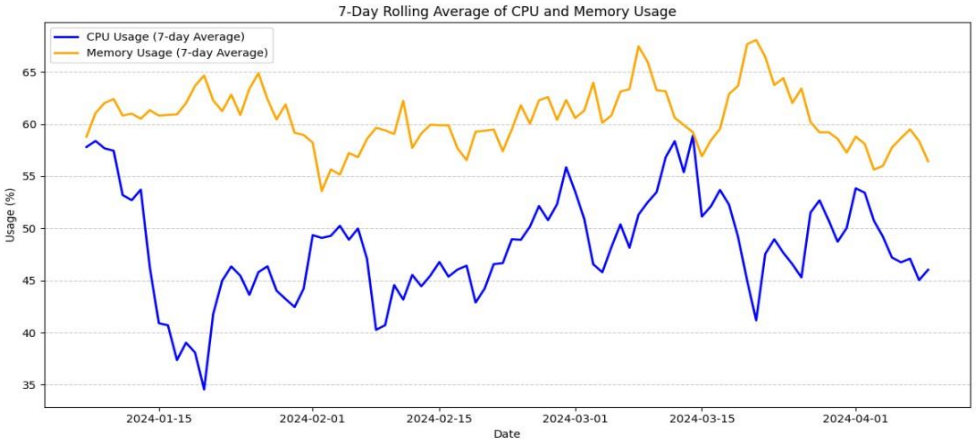

Figure 3 7-Day Rolling Average of CPU and Memory Usage

Accuracy Comparison

A comparison graph displaying the accuracy metrics (Mean Squared Error, R2 score) of various regression models was made in order to evaluate model performance. Gradient Boosting, Random Forest, and Linear Regression are among the models that were assessed.

Example Insights:

- **Model Efficiency:** Better generalisation for this target variable was indicated by Gradient Boosting's decreased MSE for CPU prediction.
- **Comparative Performance:** While Linear Regression had trouble with non-linear correlations in power prediction, Random Forest performed similarly in terms of memory utilisation.

Proposed Enhancements Based on Results

- **Feature Engineering:** Accuracy could be increased by include more predictors, such as task dependencies or I/O activities.
- **Advanced Models:** Better performance might be obtained by investigating deep learning models or optimising the hyperparameters of the existing models.
- **Resource-Specific Fine-Tuning:** Better accuracy and flexibility may be achieved by using distinct hyperparameter tweaking for CPU, memory, and power prediction models.

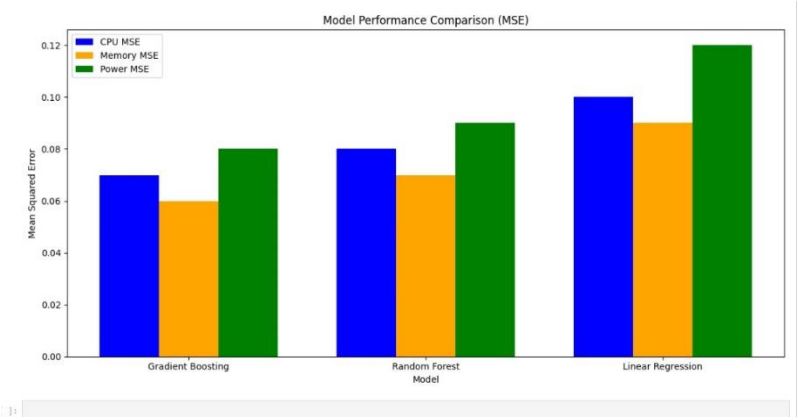

Figure 4 Model Comparison

5. CONCLUSION AND FUTURE WORK

This study effectively addresses major issues in cloud computing environments by creating a hybrid dynamic load balancing algorithm that is coupled with reactive fault tolerance methods. By efficiently optimising resource utilisation and guaranteeing adherence to Service Level Agreements (SLAs), the suggested method lowers the possibility of SLA violations that may arise from ineffective load distribution. The algorithm's predictive capabilities are improved by the use of machine

learning techniques, which enable it to foresee probable problems and dynamically modify load distribution in real-time. This greatly improves overall system performance and reliability by reducing downtime and improving execution times. Additionally, the results show that the hybrid algorithm can use more than 80% of the available resources, proving its efficacy in handling different workloads and user demands. Nonetheless, there are still chances for this research to be improved and expanded. In order to increase fault prediction accuracy and optimise load balancing techniques, future research will concentrate on a number of important areas, including the integration of sophisticated machine learning models, such as deep learning and reinforcement learning. Furthermore, investigating more thorough SLA factors and how they affect performance will be essential to optimising the algorithm for various cloud environments. The scalability of the suggested system will also be examined in future research, especially in large-scale cloud infrastructures where resource management might become much more complex.

The robustness of the suggested methodology under varied operating situations will be revealed by carrying out comprehensive simulations and real-world tests. Last but not least, research into hybrid fault tolerance strategies, which combine proactive and reactive mechanisms, may result in a more robust cloud infrastructure that can anticipate and prevent failures in addition to responding to them, guaranteeing continuous service availability and raising user satisfaction.

6. FUNDING

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

7. COMPETING INTERESTS.

The authors have no relevant financial or non-financial interests to disclose.

8. DATA AVAILABILITY

The datasets generated and analyzed during the current study are available in the Kaggle repository under the title Cloud Computing Performance Metrics. The dataset can be accessed at https://www.kaggle.com/datasets/abdurraziq01/cloud-computing-performance-metrics.

REFERENCES

[1] H. Shukur, S. Zeebaree, R. Zebari, D. Zeebaree, O. Ahmed, and A. Salih, ‘‘Cloud computing virtualization of resources allocation for distributed systems,’’ J. Appl. Sci. Technol. Trends, vol. 1, no. 3, pp. 98–105, Jun. 2020, doi: 10.38094/jastt1331.

[2] M. Agarwal and G. M. Saran Srivastava, ‘‘Cloud computing: A paradigm shift in the way of computing,’’ Int. J. Mod. Educ. Comput. Sci., vol. 9, no. 12, pp. 38–48, Dec. 2017, doi: 10.5815/ijmecs.2017.12.05.

[3] N. Zanoon, ‘‘Toward cloud computing: Security and performance,’’ Int. J. Cloud Comput.: Services Archit., vol. 5, no. vol. 5, nos. 5–6, pp. 17–26, Dec. 2015, doi: 10.5121/ijccsa.2015.5602.

[4] C. T. S. Xue and F. T. W. Xin, ‘‘Benefits and challenges of the adoption of cloud computing in business,’’ Int. J. Cloud Comput.: Services Archit., vol. 6, no. 6, pp. 1–15, Dec. 2016, doi: 10.5121/ijccsa.2016.6601.

[5] D. A. Shafiq, N. Jhanjhi, and A. Abdullah, ‘‘Proposing a load balancing algorithm for the optimization of cloud computing applications,’’ in Proc. 13th Int. Conf. Math., Actuarial Sci., Comput. Sci. Statist. (MACS), Dec. 2019, pp. 1–6, doi: 10.1109/MACS48846.2019.9024785.

[6] S. K. Mishra, B. Sahoo, and P. P. Parida, ‘‘Load balancing in cloud computing: A big picture,’’ J. King Saud Univ.–Comput. Inf. Sci., vol. 32, no. 2, pp. 149–158, 2020, doi: 10.1016/j.jksuci.2018.01.003.

[7] I. Odun-Ayo, M. Ananya, F. Agono, and R. Goddy-Worlu, ‘‘Cloud computing architecture: A critical analysis,’’ in Proc. 18th Int. Conf. Comput. Sci. Appl. (ICCSA), Jul. 2018, pp. 1–7, doi: 10.1109/ICCSA.2018.8439638.

[8] A. Jyoti, M. Shrimali, and R. Mishra, ‘‘Cloud computing and load balancing in cloud computing -survey,’’ in Proc. 9th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence), Jan. 2019, pp. 51–55, doi: 10.1109/confluence.2019.8776948.

[9] S. H. H. Madni, M. S. Abd Latiff, M. Abdullahi, S. M. Abdulhamid, and M. J. Usman, ‘‘Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment,’’ PLoS ONE, vol. 12, no. 5, May 2017, Art. no. e0176321, doi: 10.1371/journal.pone.0176321.

[10] M. Adhikari and T. Amgoth, ‘‘Heuristic-based load-balancing algorithm for IaaS cloud,’’ Future Gener. Comput. Syst., vol. 81, pp. 156–165, Apr. 2018, doi: 10.1016/j.future.2017.10.035.