

Load Balancing in Financial Cloud with Dynamic Task Prioritization: An Efficient Security Model Perspective

Dipti N. Kashyap^{1*}, Dr.T.A. Madankar², Yoghesh Dharangutti³,
Dr. Sumitra Padmanabhan⁴, R. Hannah Jessie Rani⁵, and Divya Paikaray⁶

^{1*}Assistant Professor, Department of Mechanical Engineering, Yeshwantrao Chavan College of Engineering, Nagpur, Maharashtra, India. diptikashyap22@gmail.com, <https://orcid.org/0000-0003-4364-3121>

²Assistant Professor, Mechanical Engineering, Ramdeobaba University, RBU, Nagpur, Maharashtra, India. madankarta@rknc.edu, <https://orcid.org/0009-0009-9715-8149>

³Assistant Professor, Symbiosis Centre for Advanced Legal Studies and Research (SCALSAR); Symbiosis Law School, Pune (SLS-P), Symbiosis International Deemed University, India. yogesh.d@symlaw.ac.in, <https://orcid.org/0000-0002-3437-6626>

⁴Associate Professor, Department of uGDX, ATLAS SkillTech University, Mumbai, Maharashtra, India. sumitra.padmanabhan@atlasuniversity.edu.in, <https://orcid.org/0000-0003-4846-080X>

⁵Assistant Professor, Department of Electrical and Electronics Engineering, Faculty of Engineering and Technology, Jain (Deemed-to-be University), Bangalore, Karnataka, India. jr.hannah@jainuniversity.ac.in, <https://orcid.org/0000-0002-5449-104X>

⁶Assistant Professor, Department of Computer Science & IT, ARKA Jain University, Jamshedpur, Jharkhand, India. divya.p@arkajainuniversity.ac.in, <https://orcid.org/0000-0001-7886-1538>

Received: April 18, 2025; Revised: June 02, 2025; Accepted: July 21, 2025; Published: August 30, 2025

Abstract

The scheduling and distribution of tasks is one of the biggest problems with cloud computing, a platform that is becoming more and more popular for everyday use and financial applications. However, one of the main problems in the financial segment remains to be cloud security. Several studies have demonstrated that the financial cloud load balancing system manages the arrangement of n tasks in the process flow on cloud devices, which is critical to the effectiveness of the system. To research load balancing and dynamic task prioritization in financial cloud systems, a novel adaptive weighted round robin based versatile random forest (AWRR-VRF) strategy was suggested in this research. The suggested strategy uses the versatile random forest (VRF) method for dynamic task prioritization depending on security requirements and the adaptive weighted round robin (AWRR) approach for effective load balancing in the financial cloud. To categorize the task-oriented priority of requests and enable efficient task performance, the VRF is implemented based on user behavior patterns. Based on the suggested methodology, this research is carried out using the Python program and performance is examined in terms of CPU utilization (0.0100), energy consumption (62.675), task prioritization (77,600) and optimized memory usage (5102.50) measures. Task security is enhanced to minimize security threats and maximize the protection of financial data. By

Journal of Internet Services and Information Security (JISIS), volume: 15, number: 3 (August), pp. 149-159.
DOI: 10.58346/JISIS.2025.I3.010

*Corresponding author: Assistant Professor, Department of Mechanical Engineering, Yeshwantrao Chavan College of Engineering, Nagpur, Maharashtra, India.

using the experimental assessment, this research determined that the suggested AWRR-VRF technique maximizes the financial cloud systems security and performance components.

Keywords: Financial Cloud System, Security, Load Balancing, Task Prioritization, Adaptive Weighted Round Robin-Based Versatile Random Forest (AWRR-VRF).

1 Introduction

In recent years, the financial sector has seen an enormous move toward cloud computing to address increasing requirements for scalability, agility and cost efficiency. (Shahid et al., 2020). Cloud computing offers a plethora of benefits to financial institutions, including on-demand resource provisioning, elastic scalability and reduced infrastructure overhead. However, the migration of financial services to the cloud introduced new challenges, particularly in terms of ensuring security (Costa et al., 2010), reliability and performance (Jyoti & Shrimali, 2020).

The phrase "cloud computing" describes the on-demand provision of computer resources, such as processing power, storage and software. As a result of technological innovation, a wide range of industries have begun to utilize cloud computing apps to enhance and streamline their operations (Dezhabad & Sharifian, 2018; Sulfath et al., 2025). Those programmed can be discovered at any time from various geographical locations. It supplied an extensive variety of services in several industries, including social networking, education, health care management, data storage, entertainment and more. A privately owned cloud is a firm or a company's internal information center that is not open to the general public, whereas a freely accessible cloud is made available to others on a pay-as-you-go basis

Task scheduling was used for planning tasks for optimal resource utilization by allocating specified tasks to certain tools at a specific time (Smihunova et al., 2024). Tasks are computational activities that can need a variety of processing abilities and resource requirements, such as CPU, memory, a lot of nodes in a network bandwidth and so on. The task scheduling conundrum encompassed diverse criteria per task, including priority, deadlines and execution estimated, catering to two user categories: cloud providers and consumers. Consumers aim for varied-scale problem-solving, while providers offer resources for task execution (Habeeb & Kazaz, 2023). Effective scheduling benefits consumers through resource aggregation and providers through optimized utilization, crucial for shared resource scenarios (Nezami et al., 2021).

Task planning and resource management allow suppliers of cloud services to maximize revenue and resource utilization (Gupta et al., 2022). The planning and distribution of resources seem to be substantial impediments to the efficient use of resources in the cloud. Their limitation of inefficient scheduling has prompted academics to investigate job scheduling in cloud computing. The primary assumption of task scheduling was to arrange tasks so as to minimize time loss and improve performance. Inadequate task scheduling in systems can lead to extended wait times and prioritize less critical tasks, potentially causing starvation. Effective scheduling strategies consider task characteristics, size, execution time and resource availability to prioritize tasks and make scheduling decisions efficiently (Rjoub et al., 2021).

To sum up, dynamic task prioritization combined with load balancing in financial clouds performed a strong statement to the dispute of resource optimization with safe, effective operation (Teoh et al., 2021). Financial institutions can weaken security risks (Sato et al., 2019), improve regulatory compliance and prevent sensitive financial data by gathering advanced security measures into the load-balancing process (Hosseini et al., 2023). Their paper offered a thorough security model that initiated the particular needs and drawbacks of load balancing in financial cloud environments, adding to the

modifying region of cloud computing in the financial sector. Research introduce a novel AWRR-VRF strategy that was suggested in this research. The suggested approach uses the VRF method for dynamic task prioritization depending on security requirements and the adaptive weighted round robin (AWRR) approach for effective load balancing in the financial cloud.

2 Related Work

Aakisetti et al., 2024 addressed Dynamic Priority Task-Based Scheduling (DPTS) as a factious cloud scheduling method that alter corrected task preference and organize arrangement to optimize resource utilization, enhancing system performance and assure smooth task performance. (Ding et al., 2020) suggested a Q-learning-based task-planning structure for environmentally friendly cloud computing (QEEC), engaging a centrally located mission coordinator and a Q-learning-based scheduler to reduce task response times while increasing CPU utilization. Simulation trials have verified QEEC's ability to lower the average task reaction time. (Lin et al., 2023) investigated the neuronal signatures and performance effects of gait-prioritization and manual-prioritization strategies for Parkinson's patients while double-task strolling. People (N is the number of = 34) were asked to walk with two interlocking rings, either prioritizing large strides (GP approach) or separating the rings (MP methodology). Movement criteria and circle contact time were assessed, whereas the scalp electroencephalogram was recorded (Udayakumar et al., 2023).

Torbunova et al., 2024 proposed a test case prioritization schema that combined static and dynamic prioritization algorithms. It evaluated three industrial datasets and found that the dynamic prioritization algorithm improved static algorithm performance and reduced it when static scheduling was near optimal. (Wrightson et al., 2020) evaluated the dual-task impact on loco-motor performance throughout simulator and over ground walking and discovered that prioritizing cognitive tasks enhanced walking performance while prioritizing walking tasks decreased it. (Liu et al., 2022) introduced Auto-Lambda, an automated weighting framework for learning task relationships in multi-task learning. It optimized task computed and discovered interesting learning behavior, achieving state-of-the-art performance in computer vision and robotics problems. (Leang et al., 2020) tested nine task-loading schemes on three vehicle datasets and proposed an alternative approach for trustworthy network training that outperformed the current method.

3 Methodology

The proposed system makes use of the adaptive weighted round robin (AWRR) technique for efficient load balancing in the financial cloud and the versatile random forest (VRF) method for dynamic job prioritization based on security criteria.

3.1 Adaptive Weighted Round Robin

The RR load balancing technique that research proposed in this paper was improved upon by the Adaptive WRR load balancing technique. Unplanned to RR Algorithm, by managing the dispersion of virtual machine (VM) loads in the network, this algorithm raises the bar for weighted robin procedures. The primary flaw in the RR load balancing method was that it assigned the correct virtual machine to each incoming request without considering the magnitude of the user's request. It only uses rotational decision-making to determine the subsequent VM. The weighted RR load balancing algorithm's suggested technique contemplates the workload of VMs, as well as the task's duration and processing capability, to choose which VM should be assigned to the subsequent task. The static technique of the

Adaptive WRR load balancing algorithms assigns the VMs according to their processing capabilities. The dynamic loads balancing component, or at run time, uses the load on each VM in addition to the capacity statistics to determine which VM is best for assigning the work. Tasks can occasionally take longer to complete than anticipated during system runtime because of the execution of multiple cycles of the same program using intricate runtime data. The balancer is in charge of determining which VMs are overloaded, underutilized or both. The load balancer won't allow any job migrations between VMs if they're all engaged. It will transfer the task from the overloaded VMs to the non-occupied VM if it discovers an idle VM.

When a task is finished on a virtual computer, the load balancer checks the load on those machines. To save the burden on the VM, it never evaluates VMs that load independently at a moment. According to the task's waiting time, the Adaptive WRR load balancing algorithm selects which VM to map incoming requests to or assign tasks to. Finding the tasks' waiting times is the first stage in this method.

$$\text{Waiting Time} = \sum_{M=0}^M K * S \quad (1)$$

Where,

M Amount of Cloudlet

K Length of cloudlet

S Execution time of VM

The duration of an incoming work determines how long the waiting period is. Every task's cloudlet duration is represented in an encrypted structure. The preparing duration of every VM is determined in Equation (1) above by multiplying the Cloudlet duration by the processing time. Equation (1) is utilized to determine the VM waiting period upon receiving a request of length K .

Equation (2) uses Cloudlet dimensions as a basis for calculating execution time. Research divided the Cloudlet's dimension by the VM capabilities to get the processing time of the incoming position.

The units of measurement for T are MI (million instructions), while IPS (instructions per second) is used to quantify VM capacity.

$$\text{Execution Time} = \frac{T}{VM_p} \quad (2)$$

Where,

T Cloudlet dimension

VM_p Virtual Machines Power

Once the load balancer has collected the task wait times from each of the virtual computers it has built, it starts to prepare itself. The process of mapping involves selecting an assignment from VMs with the longest waiting times, identifying the most suitable VM for the task from each VM list and allocating the incoming activity to the selected VMs according to the waiting time.

Rearranging the order is how load balancing works and the waiting time is determined by the VM's load.

3.2 Versatile Random Forest for Text Classification

The random forest method was a popular ensemble learning technique known for its reliability and effectiveness in predictive modeling across multiple domains. In this section, feature weighting for

subspace sampling and tree selection. By combining this approach, the novel and Versatile Forest randomization algorithm is described.

3.2.1 Feature Weighting Method

The following section discusses the attribute-weighted method for substructure samples in random forest settings. Consider the multidimensional feature space. $\{B_1, B_2, \dots, B_N\}$. They showed how to calculate weights $\{x_1, x_2, \dots, x_N\}$ for every element in area. The weight was utilized in a versatile technique to generate decision trees in random forests.

- **Calculate Feature Weights**

To determine feature weight, they analyzed the relationship between every input attribute A and the category feature Z . A high weight suggested a correlation between class labels in training data and feature B values. As a result, B correctly predicts the class names for fresh items. It used a two-sample t-test to calculate feature weight, which is appropriate for double-class data. To deal with multi-class circumstances, they recommend employing the chi-square algorithms to calculate the feature weight.

For the given the class feature Z which has r distinct values or classes, denoted as $z_i (i = 1, \dots, r)$ and the feature B can take o values, denoted by $b_j (for j = 1, \dots, o)$. If B is numerical, then it is a supervised discretization method. Conversely, let C be a data set consisting of $\sum_{i=1}^r \sum_{j=1}^o \lambda_{ji}$ data samples. The number of samples in C where $B = b_j$ and $Z = z_i$ is denoted by λ_{ij} . Then all these λ_{ji} form a contingency of B and Z are shown. It gives the contingency table of feature B and the class feature Z of a data set C and the chi-square statistic-based correlation is computed as:

$$Corr(B, Z) = \sum_{j=1}^o \sum_{i=1}^r \frac{(\lambda_{ji} - s_{ji})^2}{s_{ji}} \quad (3)$$

Given a contingency table where λ_{ji} it represents the observed frequency and s_{ji} denotes the expected frequency, we can express the computation of s_{ji} as follows:

$$s_{ji} = \frac{\sum_{i=1}^r \lambda_{ji} \times \sum_{j=1}^o \lambda_{ji}}{\sum_{i=1}^r \sum_{j=1}^o \lambda_{ji}} \quad (4)$$

- **The Normalized Weights**

In practical scenarios, feature weights are normalized to facilitate feature subspace sampling. Let's consider the correlation between a feature B_j and the class label feature Z , denoted as $Corr(B, Z)$ for $1 \dots, j = 1, \dots, M$. Research defines,

$$x_j = \frac{\sqrt{corr(B_j, Z)}}{\sum_{j=1}^M \sqrt{corr(B_j, Z)}} \quad (5)$$

The extraction of square root of the connection is a common technique for smoothing. It can be B_j . This weight data to perform will be used in feature subspace selecting when designing our algorithm.

3.3 Proposed Technique

This new approach combines the strengths of adaptive weighted round robin (AWRR) and versatile random forest (VRF) algorithms to achieve load balancing and task prioritization. AWRR distributes incoming requests accordingly by server load dynamic based on current demand. It is important for load

balancing in complex systems because it ensures equal distribution and maximizes resource consumption. This results in better resource utilization, reduced critical response times and improved overall system reliability and performance in economical cloud environments with dynamic critical infrastructure optimized prioritization of versatile forest (VRF) using ensemble learning. It supports a variety of input formats and provides data models for transfer by connecting various decision trees. Customer service queues and resource allocation in distributed computing systems are two examples of applications that benefit from VRF's robust performance and flexibility in task priorities. VRF also functions well with different settings.

4 Results

Research implemented our approach in Python (v1.8) and the system configuration includes Pytorch 1.14, compatible with Python 1.8, on a Windows 11 OS. The suggested technique is examined in regards to balanced CPU utilization, energy consumption, task prioritization and optimized memory usage compared with the existing approaches, which are "Reinforcement Learning (RL), Lyrebird Optimization Algorithm (LOA), Falcon Optimization Algorithm (FOA)" (Khan, 2024).

Energy consumption matrices improve total system efficiency and cost-effectiveness by enabling continuous evaluation and energy usage optimization. Energy Consumption comparison is displayed in Figure.1. In comparison, the performance of the existing techniques RL, LOA and FOA was 96.980, 78.731 and 86.984, while our suggested solution AWRR-VRF had 62.675. The outcomes demonstrate that our suggested approach has lower energy consumption in comparison with the existing methods.

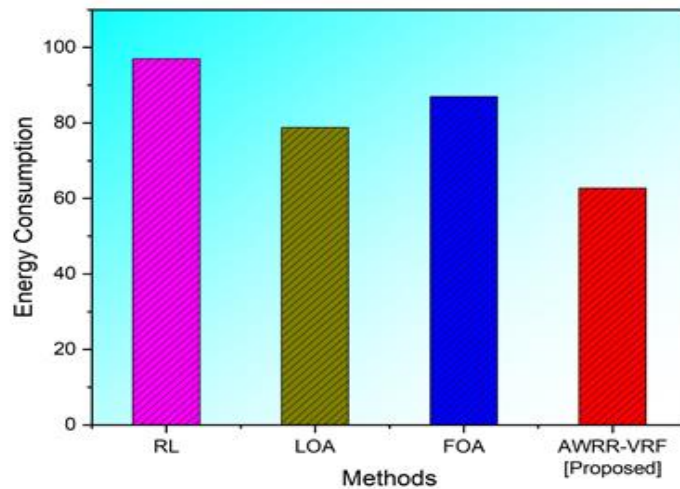


Figure 1: Output of Energy Consumption

This assists financial cloud environments allocate resources more effectively, reduce bottlenecks and maximize system performance. Figure 2 shows the balanced CPU utilization comparison and the performance of the existing technique RF, LOA and FOA, which was 0.0125, 0.0609 and 0.0362, while our proposed method AWRR-VRF had 0.0100. The results show that, in comparison to the existing approaches, our proposed technique has a substantially lower balanced CPU utilization.

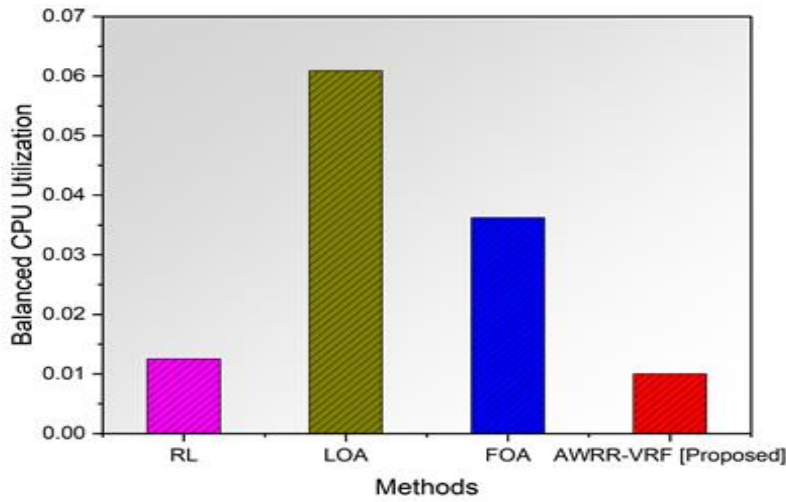


Figure 2: Output of Balanced CPU Utilization

By dynamically prioritizing processes, maintaining balanced workloads and reducing memory expenses, optimized memory utilization matrices effectively allocate resources, improving total system performance and scalability. Figure.3 shows the Optimized Memory Usage comparison and the performance of the existing technique RL, LOA and FOA, was 5721.10, 6943.92 and 7098.32, while our proposed method AWRR-VRF had 5102.50. The results show that, in comparison to the existing approaches, our proposed technique has a substantially lower optimized memory usage.

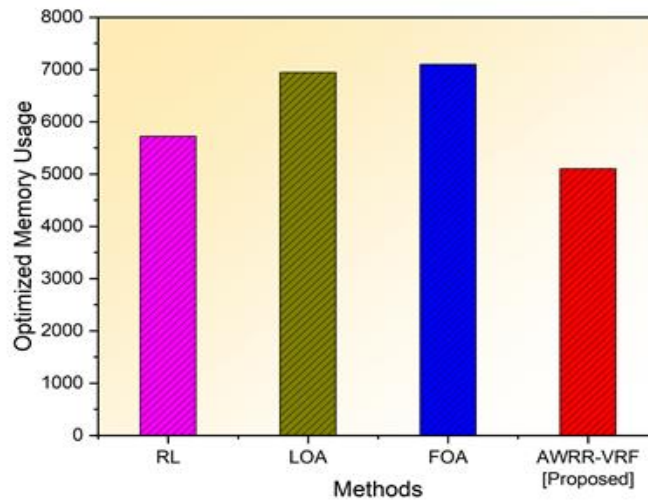


Figure 3: Output of Optimized Memory Usage

They provide direction for load balancing selections, guaranteeing that important tasks have the proper focus and resources to support effective financial cloud operations. Figure.4 shows the Task prioritization comparison and the performance of the existing technique RL, LOA and FOA, was 52,750, 53,954 and 66,598, while our proposed method AWRR-VRF had 77,600. Table 1 shows the overall result comparison. The outcomes demonstrate that our proposed approach outperforms the current methods by a substantial margin in terms of task prioritization.

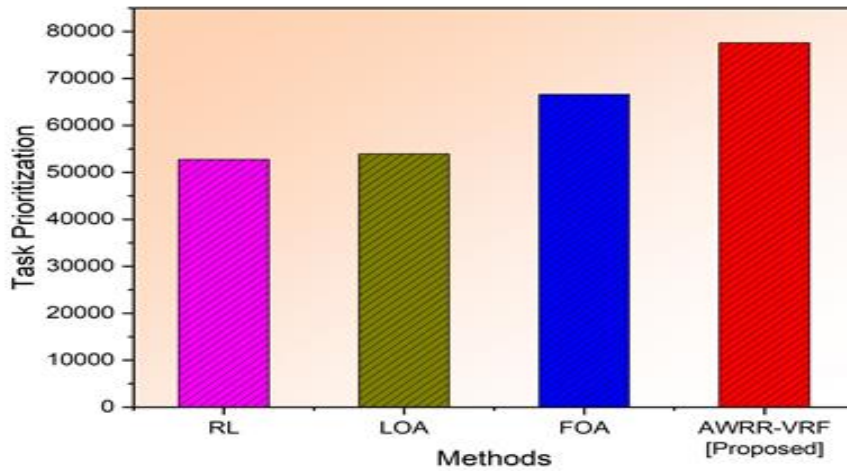


Figure 4: Output of Task Prioritization

Table 1: Overall Result Comparison

Methods	Energy Consumption	Balanced CPU Utilization	Optimized Memory Usage	Task Prioritization
RL	96.980	0.0125	5721.10	52,750
LOA	78.731	0.0609	6943.92	53,954
FOA	86.984	0.0362	7098.32	66,598
AWRR-VRF	62.675	0.0100	5102.50	77,600

5 Conclusion

Research offered a tactical solution, the AWRR-VRF, to deal with the problems associated with dynamic task prioritization and load balancing in financial cloud networks. The research emphasized how important task dispersion and planning are to cloud computing, especially in the financial industry where security considerations are major priority. The AWRR-VRF strategy integrates the AWRR methodology for efficient load balancing and the VRF technique for dynamic job prioritization based on security needs. The intention is to maximize system efficiency while increasing security measures. The evaluation of the experiment showed that the new method efficiently increases task efficiency, reduces security risks and optimizes the safety of financial data in cloud systems. Performance is evaluated in terms of balanced CPU utilization (0.0100), energy consumption (62.675), task prioritization (77,600) and optimized memory usage (5102.50) utilizing the Python program in accordance with the recommended methodology. Due to the fixed weighting scheme, AWRR-VRF may not be as effective in handling abrupt workload increases in financial clouds with dynamic task prioritization, which could result in bottlenecks in resource distribution. Future developments for AWRR-VRF in financial cloud load balancing include the incorporation of AI algorithms for the prediction of workload in real-time, the improvement of task prioritization according to transaction urgency and the augmentation of adaptability to dynamic market conditions, all of which guarantee optimal resource utilization and enhanced system performance.

References

- [1] Aakisetti, R. S. K., Ganta, V., Yellamma, P., Siram, C., Gampa, S. H., & Rao, K. B. (2024). Dynamic priority scheduling algorithms for flexible task management in cloud computing. *Int. J. Intell. Syst. Appl. Eng*, 12(13), 246-256.

- [2] Costa, G., Lazouski, A., Martinelli, F., Matteucci, I., Issarny, V., Saadi, R., ... & Massacci, F. (2010). Security-by-Contract-with-Trust for mobile devices. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 1(4), 75-91.
- [3] Dezhabad, N., & Sharifian, S. (2018). Learning-based dynamic scalable load-balanced firewall as a service in network function-virtualized cloud computing environments. *The Journal of Supercomputing*, 74(7), 3329-3358. <https://doi.org/10.1007/s11227-018-2387-5>
- [4] Ding, D., Fan, X., Zhao, Y., Kang, K., Yin, Q., & Zeng, J. (2020). Q-learning based dynamic task scheduling for energy-efficient cloud computing. *Future Generation Computer Systems*, 108, 361-371. <https://doi.org/10.1016/j.future.2020.02.018>
- [5] Gupta, S., Iyer, S., Agarwal, G., Manoharan, P., Algarni, A. D., Aldehim, G., & Raahemifar, K. (2022). Efficient prioritization and processor selection schemes for heft algorithm: A makespan optimizer for task scheduling in cloud environment. *Electronics*, 11(16), 2557. <https://doi.org/10.3390/electronics11162557>
- [6] Habeeb, A. A., & Kazaz, Q. N. N. (2023). Bayesian and Classical Semi-parametric Estimation of the Balanced Longitudinal Data Model. *International Academic Journal of Social Sciences*, 10(2), 25-38. <https://doi.org/10.9756/IAJSS/V10I2/IAJSS1010>
- [7] Hosseini, E., Nickray, M., & Ghanbari, S. (2023). Energy-efficient scheduling based on task prioritization in mobile fog computing. *Computing*, 105, 187-215. <https://doi.org/10.1007/s00607-022-01108-y>
- [8] Jyoti, A., & Shrimali, M. (2020). Dynamic provisioning of resources based on load balancing and service broker policy in cloud computing. *Cluster Computing*, 23(1), 377-395. <https://doi.org/10.1007/s10586-019-02928-y>
- [9] Khan, A. R. (2024). Dynamic load balancing in cloud computing: optimized RL-based clustering with multi-objective optimized task scheduling. *Processes*, 12(3), 519. <https://doi.org/10.3390/pr12030519>
- [10] Leang, I., Sistu, G., Bürger, F., Bursuc, A., & Yogamani, S. (2020, September). Dynamic task weighting methods for multi-task networks in autonomous driving systems. In *2020 IEEE 23rd International conference on intelligent transportation systems (ITSC)* (pp. 1-8). IEEE. <https://doi.org/10.1109/ITSC45102.2020.9294676>
- [11] Lin, Y. P., Lin, I. I., Chiou, W. D., Chang, H. C., Chen, R. S., Lu, C. S., & Chang, Y. J. (2023, February). The Executive-Function-Related Cognitive–Motor Dual Task Walking Performance and Task Prioritizing Effect on People with Parkinson’s Disease. In *Healthcare* (Vol. 11, No. 4, p. 567). MDPI. <https://doi.org/10.3390/healthcare11040567>
- [12] Liu, S., James, S., Davison, A. J., & Johns, E. (2022). Auto-lambda: Disentangling dynamic task relationships. *arXiv preprint arXiv:2202.03091*.
- [13] Nezami, Z., Zamanifar, K., Djemame, K., & Pournaras, E. (2021). Decentralized edge-to-cloud load balancing: Service placement for the Internet of Things. *Ieee Access*, 9, 64983-65000. <https://doi.org/10.1109/ACCESS.2021.3074962>
- [14] Rjoub, G., Bentahar, J., Abdel Wahab, O., & Saleh Bataineh, A. (2021). Deep and reinforcement learning for automated task scheduling in large-scale cloud computing systems. *Concurrency and Computation: Practice and Experience*, 33(23), e5919. <https://doi.org/10.1002/cpe.5919>
- [15] Sato, S., Hirose, S., & Shikata, J. (2019). Sequential Aggregate MACs from Any MACs: Aggregation and Detecting Functionality. *J. Internet Serv. Inf. Secur.*, 9(1), 2-23.
- [16] Shahid, M. A., Islam, N., Alam, M. M., Su’ud, M. M., & Musa, S. (2020). A comprehensive study of load balancing approaches in the cloud computing environment and a novel fault tolerance approach. *IEEE access*, 8, 130500-130526. <https://doi.org/10.1109/ACCESS.2020.3009184>
- [17] Smihunova, O., Bohdaniuk, I., Polyakova, Y., & Yehiozarian, A. (2024). Innovative Approaches to Controlling in Agribusiness: The Role of Quality Management Systems in Sustainable Production Practices. *Archives for Technical Sciences*, 31(2), 116-130. <https://doi.org/10.70102/afts.2024.1631.116>

- [18] Sulfath, K. K., Ramakrishnan, P. R., Shareef, P. M., & Shanmugam, H. (2025). Enhancing IT Service Management in Indian IT Organizations: A Technological Integration of ISO 20000 with AI, Blockchain, Predictive Analytics, and Zero Trust Security. *Indian Journal of Information Sources and Services*, 15(1), 267-273.
<https://doi.org/10.51983/ijiss-2025.IJISS.15.1.34>
- [19] Teoh, Y. K., Gill, S. S., & Parlikad, A. K. (2021). IoT and fog-computing-based predictive maintenance model for effective asset management in Industry 4.0 using machine learning. *IEEE Internet of Things Journal*, 10(3), 2087-2094.
<https://doi.org/10.1109/JIOT.2021.3050441>
- [20] Torbunova, A., Strandberg, P. E., & Porres, I. (2024, April). Dynamic test case prioritization in industrial test result datasets. In *Proceedings of the 5th ACM/IEEE International Conference on Automation of Software Test (AST 2024)* (pp. 154-158).
<https://doi.org/10.48550/arXiv.2402.02925>
- [21] Udayakumar, R., Mahesh, B., Sathiyakala, R., Thandapani, K., Choubey, A., & Khurramov, A. & Sravanthi, J. (2023, November). An Integrated Deep Learning and Edge Computing Framework for Intelligent Energy Management in IoT-Based Smart Cities. In *2023 International Conference for Technological Engineering and its Applications in Sustainable Development (ICTEASD)* (pp. 32-38).
- [22] Wrightson, J. G., Schäfer, L., & Smeeton, N. J. (2020). Dual-task prioritization during overground and treadmill walking in healthy adults. *Gait & posture*, 75, 109-114.
<https://doi.org/10.1016/j.gaitpost.2019.08.007>

Authors Biography



Dipti N. Kashyap is an Assistant Professor in the Department of Mechanical Engineering at Yeshwantrao Chavan College of Engineering, Nagpur, Maharashtra, India. She is actively engaged in teaching and research in mechanical engineering, with interests in areas such as thermodynamics, design, and advanced manufacturing techniques. She is dedicated to academic excellence and the development of practical engineering skills among students.



Dr. T.A. Madankar is an Assistant Professor in the Department of Mechanical Engineering at Ramdeobaba University (RBU), Nagpur, Maharashtra, India. He is actively involved in teaching, research, and academic mentorship in the field of mechanical engineering. His areas of interest include thermal engineering, fluid mechanics, and engineering design. Dr. Madankar is committed to promoting innovation and academic excellence in engineering education.



Yoghesh Dharangutti is an Assistant Professor at the Symbiosis Centre for Advanced Legal Studies and Research (SCALSAR), Symbiosis Law School, Pune (SLS-P), a constituent of Symbiosis International (Deemed University), India. He is actively involved in legal education and research, with a focus on contemporary legal issues and policy studies. Dharangutti contributes to the academic and professional development of law students through teaching, mentorship, and scholarly engagement.



Dr. Sumitra Padmanabhan is an Associate Professor in the Department of uGDX at ATLAS SkillTech University, Mumbai, Maharashtra, India. She brings extensive experience in interdisciplinary education, with a focus on innovation, design thinking, and experiential learning. Dr. Padmanabhan is deeply involved in fostering creative problem-solving skills and guiding academic research in emerging fields.



R. Hannah Jessie Rani is an Assistant Professor in the Department of Electrical and Electronics Engineering, Faculty of Engineering and Technology, at Jain (Deemed-to-be University), Bangalore, Karnataka, India. Her academic work centers on electrical and electronics engineering, and she is dedicated to delivering quality education and contributing to applied research in the field. She is passionate about integrating innovation into engineering education and nurturing the next generation of technologists.



Divya Paikaray is an Assistant Professor in the Department of Computer Science & IT at ARKA JAIN University, Jamshedpur, Jharkhand, India. She is actively engaged in teaching and research in the field of computer science, with interests that include emerging technologies, programming, and data science. Divya is committed to fostering academic excellence and contributing to the development of innovative technological solutions.