International Journal of Computer and Engineering Optimization (IJCEO)
Volume 3, Issue 1, January – February (2025)

RESEARCH ARTICLE

HYBRID DEEP LEARNING BASED MODEL FOR WORKLOAD FORECASTING IN CLOUD ENVIRONMENT

Sathiya R.R [1,*] and Ahilan A[2]

[1] Department of Computer Science and Engineering, Amrita School of Computing, Coimbatore, Amrita Vishwa Vidyapeetham, India

[2] Associate Professor, Department of Electronics and Communication Engineering at PSN College of Engineering and Technology, Melathediyoor, Tamil Nadu 627451 India

[*]*Corresponding e-mail: rr_sathiya@cb.amrita.edu*

Abstract – Cloud computing is a fundamental paradigm for computing services based on the elasticity attribute, in which available resources are effectively adjusted for changing workloads over time. One of the most important challenges in such systems is the task scheduling problem, which aims to identify the optimal resource allocation to maximize performance and minimize response times. To get around these restrictions, the new Dynamic BIRCH based BIGRU model for time series prediction (DynaBit) technique is suggested for future server load prediction. Predicting time series, collecting workloads, preprocessing and clustering them, and post-processing the data are all steps in the suggested approach. The workload data will be divided according to a historical time window during the preprocessing phase. The time series data will then be clustered based on the latency classes using the Dynamic Birch algorithm. The original data is recovered through postprocessing, and the Bidirectional Gated Recurrent Unit (BIGRU) is employed in the time series prediction phase. The proposed model has been compared with previous approaches involving Parallel Algorithm, HEFT and FCFS approaches in terms of prediction accuracy by 31.9%, 18.74%, and 12.16%, respectively.

Keywords *– Workload prediction, server, Gated Recurrent Unit, Dynamic birch.*

1. INTRODUCTION

Cloud computing works on a pay for each use system where clients access the cloud services without having full knowledge of the distribution policies and hosting specifics .This reduces the amount of time needed to shop for businesses and ascertain the logical conclusions by offering worldwide on-request access to a shared pool of resources, including storage space, computer servers, and web facilities.[1-3] Customers don't have to contact the facility provider and can access these resources consistently without stress. The goal of cloud infrastructure is to give dynamic applications a user-friendly workspace [4, 5].

A cloud permits workloads to be easily installed and scaled owing to the fast provisioning of a virtual or physical machine. Multiple virtual machines (VMs) can share physical resources (CPU, memory, and bandwidth) on a single physical host in a cloud computing environment.[6] Additionally, network virtualization allows multiple VMs to share a data center's bandwidth. Since there are typically a lot of user requests, a significant challenge is to efficiently schedule user requests with minimal turnaround time for tasks related to user demands, ensuring optimal system performance [7, 8].

The work load problem is formulated using a queuing model with the objective of minimizing the overall waiting time for each task. Task priorities are assigned according to size for effective handling, and a waiting time matrix is introduced to support the scheduling framework [9, 10]. The waiting queue is optimized using a Fibonacci heap, and a parallel algorithm is suggested for both preemptive and non-preemptive scheduling, enhancing responsiveness However, integrating several different parts, like the Fibonacci heap, parallel scheduling logic, and priority assignment algorithm, adds a lot of computational overhead and makes implementation challenging [11, 12]. This can limit the system's scalability or real-time applicability in environments with limited resources or high levels of dynamicity. To overcome this issue a novel based approach has been proposed to efficiently schedule user requests with minimal turnaround time for tasks related to user demands, ensuring optimal system performance [13]. The key contributions of the developed LATS approach have been provided in the following manner.

- In the preprocessing stage, the workload data with timestamp is sliced by a history time window and given to the Dynamic Birch which efficiently clusters the workload with similar characteristics.

- Dynamic time wrapping has been applied to dynamic birch, for refining the detection of cluster centers and reducing the influence of outliers.
- The clustered output will be fine-tuned and given to the Bidirectional Gated Recurrent Unit, which predicts the time series data that enhances the long-term dependency and maintains the efficiency.
- An evaluation of the accuracy of future workload predictions was conducted based on actual requests to web servers, and the silhouette score was utilized as the metric for assessing cluster performance.

The rest of the paper is organized in the following manner. The literature review is explained in Section II. The developed DYNABIT is extensively given in Section III. The Experimental Results section is covered in Section IV. The conclusion and future work are discussed in Section V.

2. LITERATURE REVIEW

In 2023 Gad et al. [14] introduces an opposition based simulated annealing particle swarm optimizer (OSAPSO) to address PSO's premature convergence issue. The results reveal that OSAPSO beats its peers in IIOT task scheduling of cloud systems. However, Combining POBL, SA-inspired crossover, and greedy OES strategies makes OSAPSO algorithmically complex.

In 2024 Ahmed et al. [15] suggested datasets to compare scheduling algorithms, including Shortest Job First, First Come, First Served, (DVFS) and Energy Management Algorithms (EMA). The experimental findings indicate that increasing the number of virtual machines reduces Makespan. However, improved HEFT algorithms outperform the standard HEFT algorithm in terms of shorter schedule lengths for running on several virtual machines are workflow issues.

In 2024 Devi et al. [16] suggested This paper aims to introduce an optimal hybrid metaheuristic algorithm by leveraging the strengths of both the Artificial Gorilla Troops Optimizer (GTO) and the Honey Badger Algorithm (HBA) to find an approximate scheduling solution for the BoTS problem. The result shows GTOHBA achieved 8.46–30.97% makespan reduction and 8.51–33.41% energy consumption reduction against the tested metaheuristics.

In 2023 Lipsa et al. [17] suggested a parallel algorithm for task scheduling in which the priority assignment to task and building of heap is executed in parallel with respect to the non-preemptive and preemptive nature of tasks. The results proves that our proposed algorithms perform better in terms of optimizing the overall waiting time as well as the CPU time consumed.

In 2023 Hai et al. [18] suggested different HEFT algorithm versions altered to produce improved results. The result shows that the altered versions of the HEFT algorithm have a better performance than the basic HEFT algorithm regarding decreased schedule length of the workflow problems. However, an optimization problem related to this is the maximal determination of cloud computing scheduling criteria.

In 2023 Yadav et al. [19] suggested an improved & enhanced ordinal optimization technique to reduce the large search space for optimal scheduling in the minimum time to achieve the goal of minimum makespan. This proposed ordinal optimization technique and linear regression generate optimal schedules that help achieve minimum makespan.

In 2023 Banerjee et al. [20] suggested a novel method for job scheduling in cloud computing. However, algorithm demonstrated significant improvements in makespan reduction and resource utilization compared to existing scheduling algorithms, the comparison was limited to a specific set of algorithms. The results show proposed DynaBit algorithm outperforms the other considered scheduling algorithms across various evaluation metrics.

3. PROPOSED METHODOLOGY

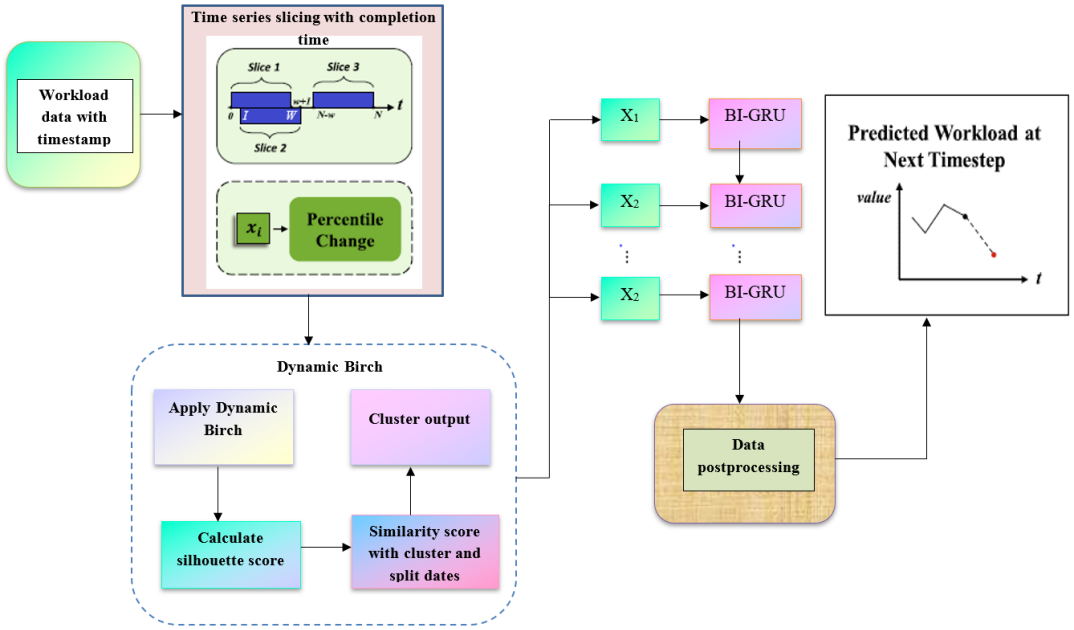

Figure 1. DynaBit technique

In this section, a novel Dynamic BIRCH based BIGRU model for time series prediction (DynaBit) algorithm has been proposed to predict future server loads. Workload collection, data preprocessing, time series forecasting using bidirectional gated recurrent units (BIGRU), and data postprocessing are all included in the suggested method. The overall block diagram for the developed methodology is given in Fig 1.

A structured forecasting approach needs to be introduced in workload forecasting because the cloud workloads exhibit complex temporal dependencies across multiple resource metrics such as CPU, memory, etc. To accurately predict future workloads, we model these dependencies using multivariate time series.

3.1. Preprocessing stage

The preprocessing stage has been divided into two steps. Initially, based on the history time window, data slicing is performed, and then the time series will be clustered.

3.2. Workload collection and data slicing

For efficient resource allocation, the DynaBit model forecasts cloud workload using workload time series data, which is collected at minute-level intervals. Because users often access the same data, workloads often exhibit recurring patterns. In order to ensure trace stability and enable effective feature mining and tuning, this study treats workload prediction as a multivariate time series problem and employs normalization techniques.

3.3. Dynamic Birch

In BIRCH, a cluster is defined by its Cluster Features (CFs), and the hierarchical structure of clusters is displayed using a CF tree. To find the cluster centroid, represented by $\{\overrightarrow{X_1}\}$ in BIRCH clustering, where $i = 1,2,\dots,N$, Equation (1) applied consecutively.

$$\overrightarrow{X_0} = \frac{\sum_1^N \overrightarrow{X_1}}{N} \quad (1)$$

By choosing the number of clusters, the processed data is separated into discrete subgroups according to specific CFs. Cluster tags are then applied to these subsets to cluster them in an energy-constrained manner.

3.3.1. Dynamic time warping (DTW) metric

DTW is selected as the distance function because storage analysis helps in detecting future workload demands. A technique employed in time series analysis to compare two temporal sequences that don't exactly match in length, velocity, or time is Dynamic Time Warping (DTW). DTW can collect temporal features with varying timing shifts. DTW modifies the correspondence between two sequences by dynamic programming concepts to detect the best path that reduces the distance between the two sequences along the path.

$$D(i,j) = \min_{\pi} \sqrt{\sum_{(x,y)\in\pi} d(i_x, j_y)^2} \quad (2)$$

Where, $\pi = [\pi_0, \dots \pi_z]$ is a path, and $d(i_x, j_y)$ is the distance.

3.3.2 Cluster fine-tuning

Initial clustering often results in poorly separated groups, fine-tuning is necessary. The framework enhances clustering by combining clusters with trend similarity above a threshold (t1), re-clustering when the proportion of outliers falls between Pmin and Pmax, classifying all dates as outliers if their proportion exceeds a maximum threshold (Pmax), keeping initial clusters if outliers are below a minimum threshold (Pmin), and moving dates with low similarity to an "outlier" cluster (t2).

3.4. BIGRU For Workload prediction

After preprocessing, the time series are passed to the Bidirectional Gated Recurrent Unit a gated recurrent unit, which is a simplified version of LSTM (Long short-term memory). Compared with LSTM, Bidirectional Gated Recurrent Unit simplifies the gating unit reduces the network parameters and is less likely to produce overfitting, and Bidirectional Gated Recurrent Unit achieves better results with the same number of iterations, so Bidirectional Gated Recurrent Unit can make the network structure simpler while maintaining the LSTM effect. At present, Bidirectional Gated Recurrent Unit has been widely used. Bidirectional Gated Recurrent Unit includes an update gate and a reset gate, which determine the retention and discarding of information respectively.

$$F_t = \sigma(T_F[R_n - 1, Y_n]) \quad (3)$$

$$S_t = \sigma(T_S[R_n - 1, Y_n]) \quad (4)$$

$$\widetilde{M_n} = \text{tanm}(T_m[S_t \odot M_n\text{-}1, Y_n]) \quad (5)$$

$$M_n = (1\text{-}F_t) \odot M_n\text{-}1 + F_t \odot \widetilde{M_n} \quad (6)$$

Where F_t is the update gate at time step t; S_t is the reset gate at time step t; $\widetilde{M_n}$ is the state of the hidden layer unit at time step t; M_n is also used as the input for the next time step; Y_n is the input at the current time step t; M_n-1 is the state of the hidden layer unit at the previous moment.

$$H_t = \vec{L} : \overleftarrow{L} \quad (7)$$

Where $\vec{L}$ is the state of gated recurrent unit for forward and $\overleftarrow{L}$ is the state of gated recurrent unit for backward. Finaly BIGRU model predicts workload

4. RESULT AND DISCUSSIONS

To assess LATS, simulation tests were performed and its performance was compared to that of the current approaches covered in Section II. Google's public workload traces, which included over 46 million activities from a variety of CPU- and memory-intensive tasks, were used in the study. The dataset, which was gathered from more than 12,500 computers, contains parameters like parent ID, time, CPU workload, job ID, number of cores, and RAM.

4.1 Clustering Results

Clusters are manually labeled according to their mean values, emphasizing the allocation and use of resources. Average memory usage remains below 50% and CPU utilization hovers around 60%, even though over 80% of the system is allocated. With CPU and memory labels connected to branches at the same clustering level, labels have a

hierarchical structure. The memory/core ratio of clusters with the label "CPU" may still be higher than that of clusters with the label "Mem."

4.2 Performance Metrics

For assessing the efficacy of the developed DYNABIT approach, we computed mean absolute percentage error (MAPE), mean absolute error (MAE), and root mean square error (RMSE). There were two separate evaluation criteria applied. The output is designated as X, while the ground truth is recorded as x.

$$RMSE = \sqrt{\frac{1}{n}(\sum_a (X_a - x_a)^2)} \quad (8)$$

$$MAE = \frac{1}{n}(\sum_a |X_a - x_a|) \quad (9)$$

$$MAPE = \frac{1}{n}\left(\sum_a \frac{|X_a - x_a|}{X_a}\right) \quad (10)$$

The suggested LATS model is evaluated by contrasting it with three current methods: FCFS, Heft and Parallel Algorithm. These represent a variety of methods and were chosen for their efficacy in workload prediction. Other approaches were disregarded because of their limited applicability and varying scopes.

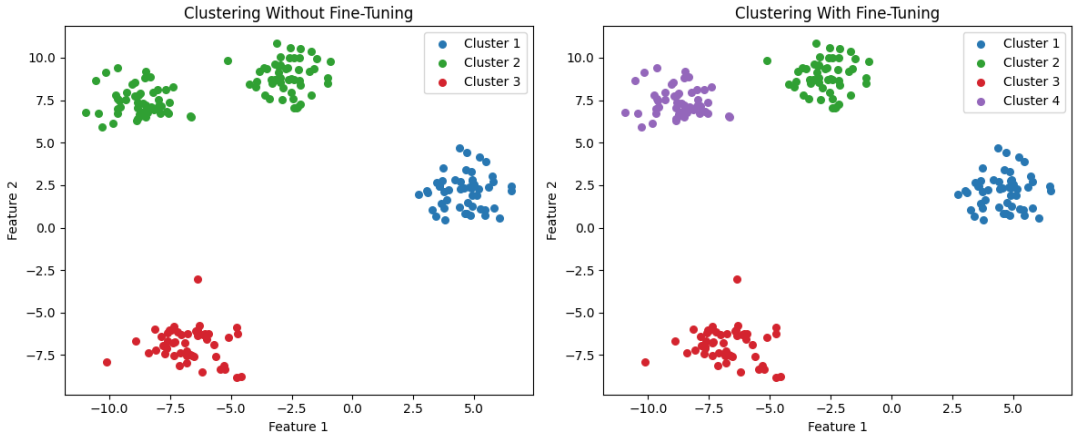

Figure 2. Clustering results (a) Clustering without fine-tuning (b) with fine-tuning

Fig 2 shows Workload clustering results Fig2(a) three overlapping clusters are produced without fine-tuning, and (b) four well-separated clusters are produced with fine-tuning. Workload differentiation and clustering accuracy are enhanced by fine-tuning.

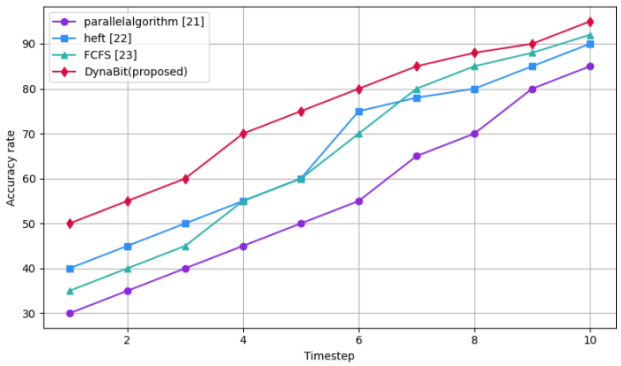

Figure 3. Accuracy rate

The prediction accuracy of the suggested LATS method, Parallel Algorithm, HEFT and FCFS is displayed in Figure 3. Because DYNABIT takes latency classes into account, it performs 31.9% better than the others, 18.74% better, and 12.16% better, respectively. Earlier approaches, on the other hand, relied on deep learning without latency awareness, which led to decreased accuracy.

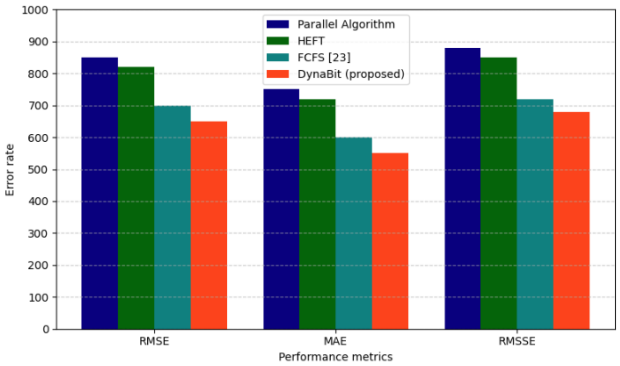

Figure 4. Comparison of performance metrics

Figure 4 compares error metrics (RMSE, MAE, RMSSE) across techniques. The proposed LATS method shows the lowest error rates due to latency-aware clustering and dynamic birch, which enhance BIGRU forecasting accuracy.

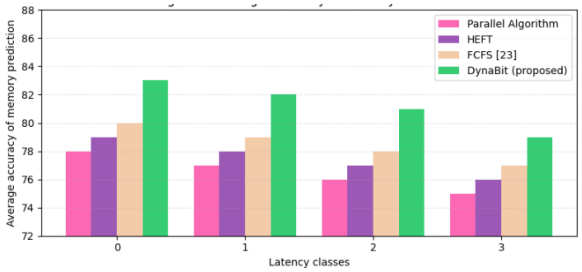

Figure 5. Average Accuracy of memory prediction

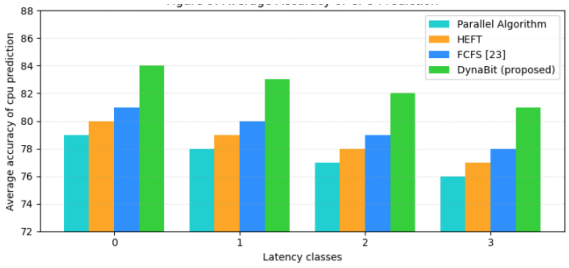

Figure 6. Average Accuracy of CPU Prediction

The suggested DYNABIT model achieves higher memory and CPU prediction accuracy across all latency classes, as seen in Figures 5 and 6. Better generalization is made possible by DTW-based clustering of related workloads. The lack of clustering in current approaches, on the other hand, results in increased errors for dynamic workloads.

5. CONCLUSION

This study suggests a Dynamic BIRCH based BIGRU model for Time series prediction (DynaBit) algorithm to predict future server loads. The developed model has been assessed utilizing real-world workload traces. When workload prediction is considered a translation problem, additional translation methods may be available. The proposed DynaBit technique is optimized for short-term predictions (one-step forecasting), which may reduce the capability of predicting long-term workloads. The numerical evaluation demonstrated that DynaBit outperformed three existing time series prediction techniques, Parallel Algorithm, HEFT and FCFS, in terms of RMSE, MAE, and MAPE. The prediction method improves significantly from the workload classification based on latency sensitivity, which is included in the proposed approach. For less latency-sensitive workloads, the proposed model outperforms Parallel Algorithm, HEFT, and FCFS approaches in terms of prediction accuracy by 31.9%, 18.74%, and 12.16%. In future work, Validating the proposed model in a real-time environment for evaluating its practical application.

CONFLICTS OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

FUNDING STATEMENT

Not applicable.

ACKNOWLEDGEMENTS

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

REFERENCES

[1] G. Zhou, W. Tian, R. Buyya, R. Xue, and L. Song, “Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions”, Artificial Intelligence Review, vol. 57, no. 5, pp.124, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] S.U. Mushtaq, S. Sheikh, and S.M. Idrees, “Next-gen cloud efficiency: fault-tolerant task scheduling with neighboring reservations for improved cloud resource utilization”, IEEE Access. 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] S.S. Mangalampalli, G.R. Karri, S.N. Mohanty, S. Ali, M.I. Khan, S. Abdullaev, and S.A. AlQahtani, “Multi-objective Prioritized Task Scheduler using improved Asynchronous advantage actor critic (a3c) algorithm in multi cloud environment”, IEEE Access, vol. 12, pp.11354-11377, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] S.K. Paul, S.K. Dhal, S.K. Majhi, A. Mahapatra, P.K. Gantayat, and S. Panda, “Optimizing task scheduling and resource utilization in cloud environment: A novel approach combining pattern search with artificial rabbit optimization,” IEEE Access, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] P. Choppara, and S. Mangalampalli, “Reliability and trust aware task scheduler for cloud-fog computing using advantage actor critic (A2C) algorithm”, IEEE Access, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] Y. Zhang, and J. Wang, “Enhanced Whale Optimization Algorithm for task scheduling in cloud computing environments,” Journal of Engineering and Applied Science, vol. 71, no. 1, p.121, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] H. Hussain, M. Zakarya, A. Ali, A.A. Khan, M.R.C. Qazani, M. Al-Bahri, and M. Haleem, “Energy Efficient Real-time Tasks Scheduling on High Performance Edge-Computing Systems using Genetic Algorithm”, IEEE Access, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] F. Yunlong, and L. Jie, “Incentive approaches for cloud computing: challenges and solutions”, Journal of Engineering and Applied Science, vol. 71, no. 1, pp. 51, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Z. Zhang, C. Xu, S. Xu, Huang, L. and J. Zhang, “Towards optimized scheduling and allocation of heterogeneous resource via graph-enhanced EPSO algorithm”, Journal of Cloud Computing, vol. 13, no. 1, pp.108, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Y. Alahmad, and A. Agarwal, “Multiple objectives dynamic VM placement for application service availability in cloud networks”, Journal of Cloud Computing, vol. 13, no. 1, pp.46, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] M. Aldossary, “Optimizing Task Offloading for Collaborative Unmanned Aerial Vehicles (UAVs) in Fog—Cloud Computing Environments”, IEEE Access, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] I.Z. Yakubu, and M. Murali, “An efficient IoT-fog-cloud resource allocation framework based on two-stage approach”. IEEE Access, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] O.L. Abraham, M.A. Ngadi, J.M. Sharif, and M.K.M. Sidik, “Task Scheduling in Cloud Environment—Techniques, Applications, and Tools: A Systematic Literature Review”, IEEE Access, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] A.G. Gad, E.H. Houssein, M. Zhou, P.N. Suganthan, and Y.M. Wazery, “Damping-assisted evolutionary swarm intelligence for industrial IoT task scheduling in cloud computing,” IEEE Internet of Things Journal, vol. 11, no. 1, pp.1698-1710, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] A. Ahmed, M. Adnan, S. Abdullah, I. Ahmad, N. Alturki, and L. Jamel, “An efficient task scheduling for cloud computing platforms using energy management algorithm: A comparative analysis of workflow execution time”, IEEE Access, vol. 12, pp. 34208-34221, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] A.G. Hussien, A. Chhabra, F.A. Hashim, and A. Pop, "A novel hybrid Artificial Gorilla Troops Optimizer with Honey Badger Algorithm for solving cloud scheduling problem", Cluster Computing, vol. 27, no. 9, pp.13093-13128, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[17] S. Lipsa, R.K. Dash, N. Ivković, and K. Cengiz, "Task scheduling in cloud computing: A priority-based heuristic approach", IEEE access, vol. 11, pp. 27111-27126, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] T. Hai, J. Zhou, D. Jawawi, D. Wang, U. Oduah, C. Biamba, and S.K. Jain, "Task scheduling in cloud environment: optimization, security prioritization and processor selection schemes", Journal of Cloud Computing, vol. 12, no. 1, pp.15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[19] M. Yadav, and A. Mishra, "An enhanced ordinal optimization with lower scheduling overhead based novel approach for task scheduling in cloud computing environment", Journal of Cloud Computing, vol. 12, no. 1, pp.8, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] P. Banerjee, S. Roy, A. Sinha, M.M. Hassan, S. Burje, A. Agrawal, A.K. Bairagi, S. Alshathri, and W. El- Shafai, "MTD-DHJS: makespan-optimized task scheduling algorithm for cloud computing with dynamic computational time prediction", IEEE Access, vol. 11, pp.105578-105618, 2023. [CrossRef] [Google Scholar] [Publisher Link]

AUTHORS

Sathiya R.R. completed her Bachelor's Degree in Information Technology from Bharathiar University, Coimbatore, Tamilnadu, India, in the year 2004 and her Master's Degree in Computer Science and Engineering from Anna University, Coimbatore, Tamilnadu, India, in the year 2010. Her areas of interest include cloud computing, Internet of Things, and Image processing. She has published technical papers in International Conferences and Journals.

Ahilan A. received Ph.D. from Anna University, India, and working as an Associate Professor in the Department of Electronics and Communication Engineering at PSN College of Engineering and Technology, India. His area of interest includes FPGA prototyping, Computer vision, the Internet of Things, Cloud Computing in Medical, biometrics, and automation applications. Served Guest editor in several journals of Elsevier, Benthom, IGI publishers. Also, have contributed original research articles in IEEE Transactions, SCI, SCIE, and Scopus indexed peer-review journals. He presented various international conference events like ASQED (Malaysia), ESREF (France). He is doing as a reviewer in IEEE Industrial Informatics, IEEE Access, Measurement, Multimedia Tools & Applications, Computer Networks, Medical systems, Computer & Electrical Engineering, neural computing and applications, Cluster Computing, IET Image Processing, and so on. He has IEEE and ISTE membership. He has worked as a Research Consultant at TCS, Bangalore, where he has guided many computer vision projects and Bluetooth Low Energy projects. Hands on programming in MATLAB, Verilog and python at various technical institutions around India.

Arrived: 04.01.2025
Accepted: 13.02.2025