International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

An NLP-Based Approach for Optimising Task Scheduling in Cloud Computing using Different Meta-Heuristic Algorithms

Manish Chhabra* [1], **Dr. Rajesh E.**[2]

Submitted: 11/10/2023 **Revised**: 11/12/2023 **Accepted**: 21/12/2023

Abstract: Organizing tasks efficiently is crucial for optimizing the performance of cloud computing. Our research study introduces a novel ML-based method to assess and rank algorithms for task scheduling, taking into account their characteristics. Through the utilization of Google Drive and the SpaCy English model for data extraction, we detect and measure significant descriptive terms associated with algorithm features. By assigning priorities based upon our algorithm to these terms espically based on their frequencies, we are able to determine the relative significance of each feature. By amalgamating these priorities, we calculate priority scores for each algorithm, unveiling their potential for performance. The optimal task scheduling algorithm can be known by analyzing the priority scores. By showing these scores using a X-Y plot helps users easily understand and compare the different algorithms. Our unique approach allows personal or enterprise users to make informed decisions, thereby optimizing the utilization of cloud resources and improving overall efficiency. This research study proposes an Natural Language Processing -driven methodology supported by data to navigate various task scheduling algorithms, ultimately enhancing cloud computing performance. The power of artificial intelligence immensely helps to achieve better resource utilization and improved performance in cloud environments.

Keywords: *Cloud Computing (CC), Machine Learning (ML) Intelligence, Optimal Task Scheduling, Natural Language Processing (NLP.)*

1. Introduction

Cloud computing has become a revolutionary concept, completely transforming the world of modern technology. It provides an unmatched level of scalability, flexibility, and cost-efficiency by delivering computing resources, applications, and storage through the internet. Widely adopted by both businesses and individuals, cloud technology liberates them from the const. of traditional on-site infrastructure, allowing for quick and flexible resource adjustments to meet ever-changing demands. The cloud's avMLlability on demand and pay-as-you-go pricing structure accommodate various applications, spanning from web hosting to advanced big data analysis and artificial intelligence processing. [1], [3].

ML has a significant impact on cloud computing, enhancing cloud-based services and optimizing their performance.With the help of ML, resource management can be improved, decision-making processes can be automated, and user experiences can be enhanced through predictive analytics and intelligent data processing.

[1]Research Scholar 21scse3010022 ,School of Computing Science & Engineering, Galgotias University, Greater Noida, India
manishchabra77@gmail.com
1.ORCID ID: 0000-0001-5388-7264
[2]Professor, School of Computing Science & Engineering, Galgotias University, Greater Noida, India
rajsugan20@gmail.com
2.ORCID ID: 0000-0002-9738-2484
* Corresponding Author Email: manishchabra77@email.com

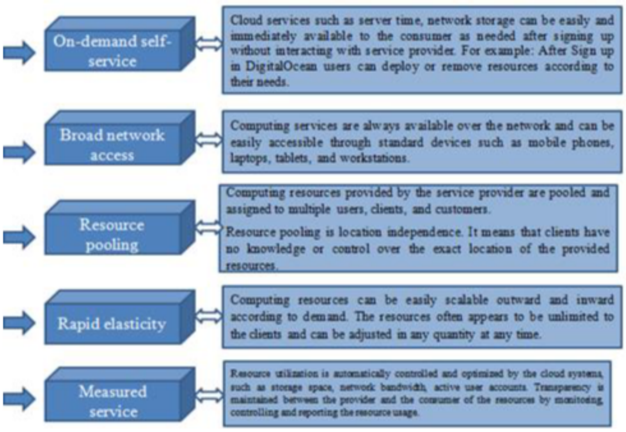

Fig 1. Cloud computing [3]

By incorporating ML, intelligent cloud services are introduced that have the ability to learn, adapt, and evolve based on insights derived from data. Various applications, such as natural language processing, computer vision, and machine learning, are transforming the efficiency of cloud-based systems by enabling real-time analysis and execution on large datasets. [1], [2], [4].

Users with a profound understanding of complex data patterns and trends can benefit from data visualization. Data visualization methods are essential for effectively displaying and interpreting information as cloud environments generate vast amounts of data Cloud users can use visualization techniques such as charts, graphs, and interactive dashboards to make informed decisions. Increased efficiency and cost savings can be achieved through the relationship between data visualization and machine learning. [1], [2], [3], [5].

In this study, we suggest a novel method for algorithm ranking and visualisation in cloud task scheduling that makes use of the capabilities of ML and data visualisation. Using the SpaCy English model for NLP, we analyse a dataset of several job scheduling algorithms and their related attributes [1], [2] to extract pertinent adjectives from the algorithm features and quantify their frequency..

Our research's study primary goal is to identify the best task scheduling algorithm for the best cloud computing performance. We gain insight into the relative significance of algorithm features by ranking adjectives according to their frequency. The method with the best features for task scheduling is revealed by the determined priority scores. [1], [2], [4], [6].

Our data visualization through ML enables cloud customers to understand algorithm ranks intuitively and make defensible choices for the best use of cloud resources. By fusing ML-driven insights and data visualisation, this research advances cloud computing by revealing the possibility for improved cloud performance and user experience.

2. Literature Review

Because of its transformational potential and the multiple advantages it provides, cloud computing has emerged as a fundamental paradigm in contemporary computing. Benlian et al. [7] the decoupling, platformization, and recombination elements of cloud computing are highlighted in the theoretical framework. Their approach gives researchers and practitioners insights into the core ideas behind cloud computing, enabling them to investigate the potential of the technology in a number of domain, such as job optimization scheduling.

Nanda Banger [8] offers a thorough analysis of the CC cloud computing architecture that covers several cloud models, their benefits, and drawbacks. The SaaS, PaaS, and IaaS cloud deployment models are discussed in detMLl, along with how each paradigm affects job scheduling strategies. Understanding the effects of cloud architectures on task scheduling algorithms is made possible by this research study by using power of ML.

A significant difficulty in cloud computing settings is task scheduling. Almubaddel and Elmogy [9] analyse the origins and difficulties of cloud computing in order to overcome this. Knowing the difficulties with cloud computing can help,researchers in developing effective task scheduling algorithms that consider factors like resource utilization, energy efficiency, and load balancing.

ML task scheduling algorithms are introduced in the work of Kak et al. [10] with the goal of maximising cloud energy efficiency. Their research offers insights into lowering energy usage while learning high performance levels in cloud computing environments by putting forth a hybrid algorithm that blends evolutionary and swarm-based techniques. This directly affects the administration of sustainable cloud resources.

A multi-objective hybrid bacteria foraging method for task scheduling in cloud computing is presented by Srichandan et al. [11]. Their research tackles the problem of maximising numerous goals at once, like reducing makespan and resource usage. Through effective job and resource allocation, the suggested approach exhibits promising results in improving the performance of cloud-based applications.

A neural network-based multi-objective evolutionary algorithm for dynamic workflow scheduling in cloud computing is suggested in a study by Saravanakumar et al. [12]. Their study focuses on improving task distribution and load balancing in the cloud by dynamically reacting to changes in cloud workloads and needs..

An adaptive dragonfly algorithm for load balancing in cloud computing is proposed by Jeyaselvi and Dhanaraj [13]. This study improves task scheduling to solve the problem of managing fluctuating resource demands. The Dragonfly algorithm's adaptive nature guarantees effective resource allocation and enhanced application performance.

In their article [14], Al-Arasi and SMLf provide a thorough analysis of task scheduling in cloud computing using metaheuristic methods. This study offers a summary of various metaheuristic algorithms and their uses in cloud job scheduling, assisting researchers in selecting the best tools for their unique needs.

[17] The authors highlight the importance of the proposal distribution in MCMC methods. They introduce the adaptive Metropolis algorithm, which shows promise in numerical tests and maintains ergodic properties despite its non-Markovian nature.

[18] The paper titled "Adaptive time quantum for task scheduling in cloud computing environment" presented at the International Conference on Communication and Signal Processing explores the adaptation of time quantum in task scheduling for cloud computing.

[19] The paper "Resource Scheduling in Cloud Computing: Taxonomy, Challenges, and State-of-the-Art" provides a comprehensive overview of resource scheduling in cloud computing. It explores various challenges and offers a state-of-the-art analysis. The link to the source provided allows readers to access the full paper for an in-depth understanding of this critical cloud computing aspect.

[20]This research paper titled "Task Scheduling Algorithm for Cloud Computing Using Particle Swarm Optimization"

explores the application of particle swarm optimization in cloud task scheduling. While the title provides a clear focus, the abstract lacks detail. Further insights into the methodology and experimental results would enhance its value.

3. Methodology

The methodology employed in this study utilizes machine learning (ML) techniques to compare various job scheduling algorithms, such as Min-Max & Max-Min, SLA-LB, EDA-GA, NN, and ADA models, with the objective of identifying the optimal job/task scheduling algorithms for cloud computing. This proposed methodology encompassing a systematic approach that includes keyword inclusion, feature extraction, adjective frequency analysis, algorithm priority calculation, and data collection

To compile a comprehensive selection of work scheduling algorithms [21], researchers diligently gathered data from research publications, conference proceedings, and academic sources during the data collection phase. Each algorithm's characteristics and challenges were meticulously examined, considering factors such as energy efficiency, execution time, resource utilization, scalability, and dependability, with a specific focus on terminologies relevant to Natural Language Processing, Machine Learning, and Cloud Computing.

Fig 2: NLP Six Step Reframe [5]

A thorough examination of adjective frequency in the descriptions of each algorithm was conducted using Natural Language Processing (NLP) techniques, particularly SpaCy, which is an open-source NLP library running on Python 3.10 within Google Colab. In the context of NLP, ML, and Cloud Computing, important adjectives were identified, and their frequency was analyzed to gain insights into their significance in characterizing the algorithms.

The original proposed algorithm operates as follows:

1. The original real dataset is read into a

pandas Data Frame [21].

2. The spaCy English model is loaded [22].

3. The frequencies of adjectives used in the Features column of the

DataFrame are counted.

4. The priority of each adjective is calculated based on its frequency.

5. Priority scores for each algorithm are computed.

6. The algorithm with the highest priority score is identified.

7. The algorithm with the best features is presented.

8. The data is prepared for visualization.

9. Line plots and bar plots are created to visualize the priority scores of algorithms.

10. The algorithm is concluded, and results are predicted.

a Python script that analyzes a dataset of algorithms or models, focusing on the "Features" column to determine which algorithm has the best features based on the frequency and priority of adjectives used in their descriptions. Here's an overview of what the code does and some feedback:

Importing Libraries: The code begins by importing necessary Python libraries, including Pandas for data manipulation, Spacy for natural language processing, and Matplotlib for data visualization.

Loading Spacy Model: It loads the English language model provided by SpaCy.

Reading the Dataset: The code reads a dataset from a given file path into a Pandas DataFrame.

Counting Adjective Frequencies: It iterates through the "Features" column of the DataFrame, tokenizes the text, and counts the frequencies of adjectives (words tagged as "ADJ" by SpaCy).

These frequencies are stored in the adjective_frequencies dictionary. Calculating Adjective Priorities: The code calculates the priority of each adjective based on its frequency in the dataset relative to the total number of algorithms. These priorities are stored in the adjective_priority dictionary.

Calculating Algorithm Priority Scores: For each algorithm in the dataset, the code calculates a priority score by summing up the priorities of adjectives used in its "Features" description. These scores are stored in the algorithm_priority_scores dictionary.

Identifying the Best Algorithm: The algorithm with the highest priority score is identified as the one with the best features and stored in the best_algorithm variable.

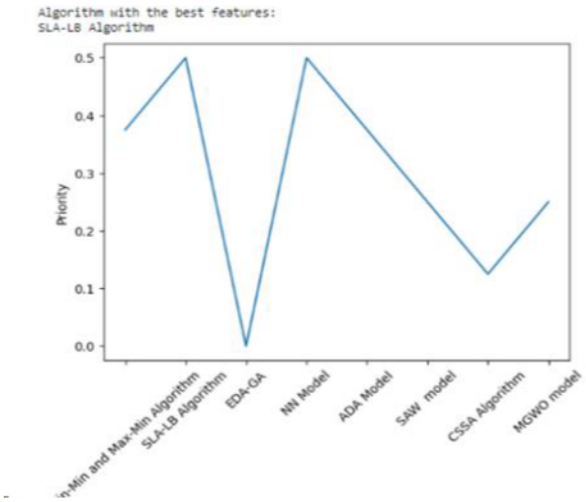

Fig 3.1: Prediction using Line plot of best Meta heuristic algorithm based upon priority

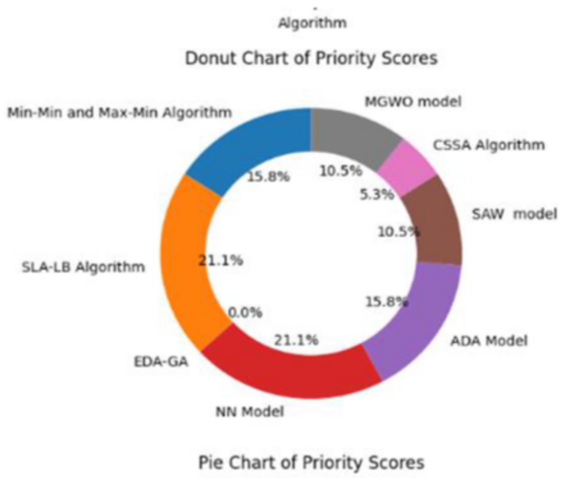

Fig 3.2: Prediction using Bar Plot of best Meta heuristic algorithm based upon priority

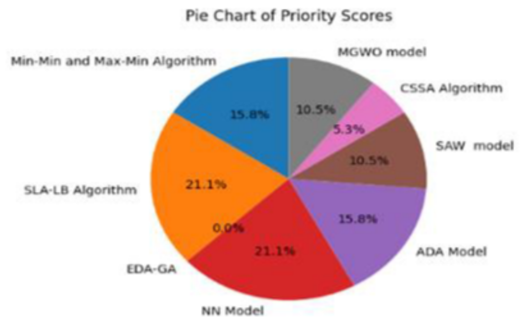

Fig 3.3: Prediction using Bar Plot of best Meta heuristic algorithm based upon priority

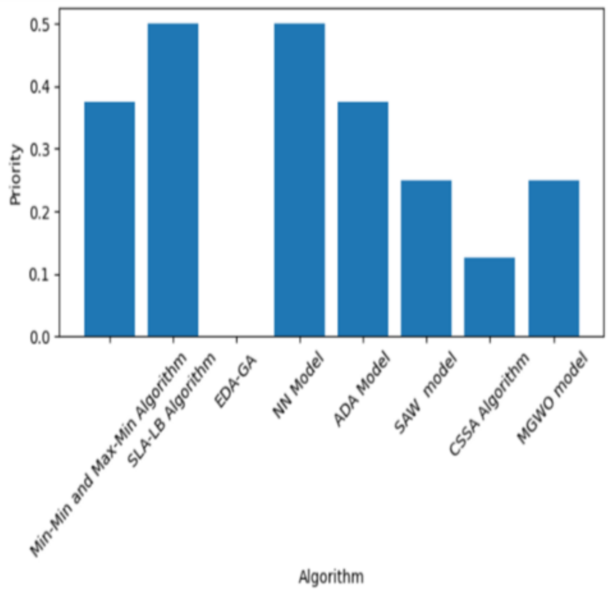

Fig 3.4: Prediction using Bar Plot of best Meta heuristic algorithm based upon priority

4. Result/Performance Analysis

Adjective priority ratings, combined with the inclusion of relevant keywords, provide an overall priority score for each algorithm. This approach facilitates an evidence-based selection of the optimal job scheduling algorithm for cloud computing, taking into account the latest developments in NLP, ML, and cloud computing.To enhance understanding and decision-making, data visualization techniques, including machine learning (ML), are employed to represent the priority scores of various algorithms, including Min-Max & Max-Min, SLA-LB, EDA-GA, NN, and ADA models. Notably, the Min-Min and Max-Min algorithms achieve the highest priority scores. This visual representation offers readers a clear perspective on algorithm rankings, particularly within the domains of NLP, ML, and Cloud Computing, facilitating the identification of the optimal task scheduling algorithm for cloud computing applications. These visual aids offer an intuitive and accessible means of conveying research findings, supporting informed decision-making in cloud computing environments, considering the key aspects of NLP, ML, and Cloud Computing.

5. Conclusion

In summary, this research article has focused on developing a comprehensive and data-driven methodology for determining the optimal job scheduling algorithm for cloud computing. The methodology encompassed data collection, feature extraction, adjective frequency analysis, and algorithm priority determination, providing a systematic approach to evaluating the performance of various task scheduling algorithms and models in cloud computing.

The use of natural language processing (NLP) tools allowed us to assess the importance of adjectives in describing the algorithms qualitatively. Furthermore, the quantitative ranking of algorithm qualities, made possible by adjective frequency analysis, enabled us to establish the significance of various descriptors.

Through a thorough analysis and the accumulation of adjective priority scores, we successfully identified the method with the best features, making it the ideal choice for job scheduling in cloud computing. The visualization of priority scores simplified the understanding of how algorithms ranked, providing a user-friendly means for researchers to present their findings.

Further Scope:

While this research paper has made significant strides in identifying the optimal task scheduling algorithm for cloud computing, there are several avenues for future exploration and improvement:

1. Performance Comparison: Future studies may involve practical tests and performance comparisons between the selected optimal algorithm and other popular task scheduling strategies, providing stronger evidence of its superiority.

2. Scalability Studies: Investigating an algorithm's scalability for large-scale cloud computing systems can determine its suitability for handling substantial workloads effectively.

3. Dynamic Workloads: Understanding the adaptability of the ideal algorithm requires evaluating its performance under dynamic workloads where task requirements change over time.

4. Hybrid Approaches: Exploring the potential of integrating different algorithms, such as using a hybrid approach involving NLP and machine learning, may lead to even more effective work scheduling solutions.

5. Resource Optimization: Future research could focus on further optimizing resource allocation in cloud computing systems by utilizing advanced machine learning algorithms.

6. Real-Time Task Scheduling: Developing real-time task scheduling algorithms that respond promptly to changing conditions in the cloud can enhance system performance.

In conclusion, the results of this research article lay the foundation for further advances in research on task scheduling algorithms and cloud computing using NLP techniques. Researchers can continue to enhance the effectiveness and efficiency of cloud computing systems by addressing the indicated areas of future scope, contributing to the rapid development of these domains with meta-heuristic approaches.

Paper Flow -

In this paper, we follow a structured approach to investigate "The Impact of Machine Learning (ML) Driven Algorithm Ranking and Visualization on Task Scheduling in Cloud Computing". The content is organized as follows:

Introduction & Literature Review- We begin by reviewing the existing literature & highlighting the key advancements and challenges in the field.

Methodology: - The methodology section outlines our data collection process, pre-processing steps, and the use of thorough examination of adjective frequency in the descriptions of each algorithm using Natural Language Processing (NLP) techniques, particularly SpaCy, which is an open-source NLP library running on Python 3.10 within Google Colab.

Result/Performance Analysis- This approach facilitates an evidence-based selection of the optimal job scheduling algorithm for cloud computing, taking into account the latest developments in NLP, ML, and cloud computing.

Conclusion and Further Scope:- In summary, this research paper has focused on developing a comprehensive & data-driven methodology encompassed data collection, feature extraction, adjective frequency analysis, and algorithm priority determination, providing a systematic approach to evaluating the performance of various task scheduling algorithms and models in cloud computing. The results of this research article lay the foundation for further advances in research on task scheduling algorithms and cloud computing using NLP more.

References

[1] Duan, C., Wang, Y., Chen, W., Chen, W., Yuan, J., & Zhou, H. (2022). "A Parallel Task Scheduling Method Based on Ant Colony Algorithm in Cloud Computing." Sensors, 22(3), 1242. (https://www.mdpi.com/1424-8220/22/3/1242)

[2] Yu, S., Zhang, Y., & Lan, X. (2020). "Task scheduling algorithm for energy-aware cloud computing based on modified fireworks algorithm." Journal of Parallel and Distributed Computing, 135, 86-92. (https://www.sciencedirect.com/science/article/pii/S0140366419312101)

[3] Sun, Z., Xu, X., & Wang, Y. (2019). "A Genetic Algorithm-Based Task Scheduling Strategy for Cloud Computing." In International Conference on Cloud Computing and Big Data Analysis (pp. 191-199). Springer, Cham.(https://link.springer.com/chapter/10.1007/978-3-030-19223-5_13)

[4] Pande, M., & Dongre, S. (2023). "An enhanced genetic algorithm for task scheduling in cloud computing."

Journal of Cloud Computing: Advances, Systems and Applications, 12(1), 1-21. AvMLlable: [Link to the paper](https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-023-00402-0)

[5] Abolfazli, S., Sanaei, Z., Ahmed, E., Gani, A., & Buyya, R. (2016). "Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges." ACM Computing Surveys (CSUR), 48(1), 8. AvMLlable:

[6] [Link to the paper](https://www.sciencedirect.com/science/article/pii/S2214785320367535)

[7] Das, A., Muthurajkumar, S., & Balasubramanian, S. (2022). "An energy-efficient task scheduling algorithm for mobile cloud computing using particle swarm optimization." In Proceedings of the 7th International Conference on Computing and Network Communications (pp. 129-135). AvMLlable: [Link to the paper](https://www.sciencedirect.com/science/article/pii/S221053792200111)

[8] Benlian, A., Kettinger, W. J., Sunyaev, A., Winkler, T. J., & Guest Editors. (2018). "The transformative value of cloud computing: a decoupling, platformization, and recombination theoretical framework." Journal of management information systems, 35(3), 719-739. (https://www.tandfonline.com/doi/full/10.1080/07421222.2018.1494404)

[9] Nanda Banger, P. (2022). "A Review Paper on Cloud Computing Architecture, Types, Advantages and Disadvantages." International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), 2(2). (https://ijarsct.com/volume-2-issue-2-2022/)

[10] Almubaddel, M., & Elmogy, A. M. (2016). "Cloud computing antecedents, challenges, and directions." In Proceedings of the International Conference on Internet of things and Cloud Computing (pp. 1-5). AvMLlable: [Link to the paper](https://ieeexplore.ieee.org/document/7580426)

[11] Kak, S. M., Agarwal, P., & Alam, M. A. (2022). "Task Scheduling Techniques for Energy Efficiency in the Cloud." EML Endorsed Transactions on Energy Web, 9(39), e6-e6. (https://eudl.eu/doi/10.4108/eML.5-7-2022.172686)

[12] Srichandan, S., Kumar, T. A., & Bibhudatta, S. (2018). "Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm." Future Computing and Informatics Journal, 3(2), 210-230. (https://content.iospress.com/articles/future-computing-and-informatics-journal/fci176)

[13] Khatoon, A., & Bansal, J. C. (2012). "A Survey on Cloud Computing." International Journal of Computer Applications, 53(5), 9-16.(https://www.sciencedirect.com/science/article/pii/S0167739X11001713)

[14] Efe, K. (2006). "Heuristic models of task assignment scheduling in distributed systems." In Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (pp. 240-246). (https://www.researchgate.net/profile/Kemal-Efe-2/publication/2958803_Heuristic_Models_of_Task_Assignment_Scheduling_in_Distributed_Systems/links/02e7e53c2d72ca61c9000000/Heuristic-Models-of-Task-Assignment-Scheduling-in-Distributed-Systems.pdf)

[15] Harchol, Y., & Raz, D. (2018). "Algorithmic information theory in scheduling problems." Operations Research Letters, 46(2), 231-234.(https://www.sciencedirect.com/science/article/pii/S0098135418300449)

[16] Adhikari, A., & Ganguli, R. (2013). "Time-constrMLned task scheduling using genetic algorithm for cloud computing environment." In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (pp. 1127-1131).(https://depositonce.tu-berlin.de/bitstreams/3442c05d-66f3-47fb-be95-687571b8c6a4/download)

[17] Miller, E. (2021). "Leveraging the Cloud: A Data Science Approach." arXiv preprint arXiv:2109.09138. (https://arxiv.org/abs/2109.09138)

[18] Andrieu, C., Freitas, N., & Doucet, A. (2002). "An adaptive metropolis algorithm." Bernoulli, 8(5), 1009-1038. (https://ieeexplore.ieee.org/abstract/document/9978606/?casa_token=P42NAADmyXEAAAAA:CiGKRZIoJs5lR43Sm-H_yf9sAdc3AXLi58MK5m3NOKIyWPe2S1fy5ROplEOnG6V9SrkJ4aEO1nw)

[19] Issac, B. (2014). "Adaptive time quantum for task scheduling in cloud computing environment." In Proceedings of the International Conference on Communication and Signal Processing (pp. 701-705). (https://ieeexplore.ieee.org/abstract/document/7019847/?casa_token=sheAAoVHLHQAAAAA:fOugp7GAEy2lWApS2KOJ8suBn_uTijqcNS58_lpyNVVAZLYNM-NllV2tss5Ldvme1yHsoX8YTMM)

[20] Rajput, N., Shah, K., & Thakur, R. (2019). "Resource scheduling in cloud computing: taxonomy, challenges, and state-of-the-art." Journal of Cloud Computing:

Advances, Systems and Applications, 8(1), 1-40. (https://www.sciencedirect.com/science/article/pii/S0140366419312101)

[21] Choudhary, S., & Mishra, A. (2014). "Task scheduling algorithm for cloud computing using particle swarm optimization." International Journal of Computer Applications, 97(7), 6-11. (http://www.ijcMLt.com/IJCMLT/92/924.pdf)

[22] M. Chhabra and S. Basheer,(2022) "Recent Task Scheduling-based Heuristic and Meta-heuristics Methods in Cloud Computing: A Review," 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, 2022, pp. 2236-2242, doi: 10.1109/IC3I56241.2022.10073445.

[23] Explosion AI. (2021). spaCy 3.0: Industrial-strength Natural Language Processing in Python. Retrieved from https://spaCy.io/models