Journal of Information Systems Engineering and Management
2025, 10(6s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Exploring Multilevel Feedback Queue Combinations and Regression-Based Time Quanta in Scheduling Algorithms

Luis William C. Meing[1], Dionisio R. Tandingan Jr.[2]

[1] Instructor, Information Technology Department, Faculty of Information Technology, College of Information Technology and Computer Science, University of the Cordilleras, Baguio, Philippines.

[2] PCpE, Computer Engineering Department, Faculty of Computer Engineering, College of Engineering and Architecture, University of the Cordilleras, Baguio, Philippines.

Email: [1]louiemeing@gmail.com, [2]drtandingan@uc-bcf.edu.ph

Orchid Id number: [1]0009-0004-0572-7302, [2]0000-0001-9958-0911

Corresponding Author: Luis William C. Meing.*

ARTICLE INFO

Received: 04 Oct 2024

Revised: 12 Dec 2024

Accepted: 20 Dec 2024

ABSTRACT

Much like day-to-day life, scheduling methods have roles in managing what to do and how much time or resources to invest. Multilevel Feedback Queues (MLFQ) are a widely used algorithm due to its capacity to handle a wide range of task types and execution times. There is increasing interest in exploring dynamic scheduling models that can adapt to live variables solved by machine learning. This study explores an approach to process scheduling by integrating linear regression into the dynamic adjustment of round robin time quanta within MLFQ systems. This paper identifies system resources that can be modelled using linear regression into dynamic round robin, introduces an algorithm using linear regression to optimize system performance in an MLFQ implementation, and explains how dynamic time quanta affect the performance of MLFQ scheduling algorithms. SJF and Round Robin are the most effective combination within the MLFQ system, while FCFS would be the last layer to ensure task completion. Average Waiting Time and Average Turnaround Time are the most appropriate metrics for evaluating MLFQ systems. The integration of linear regression into dynamic time quantum adjustments improves performance by reducing AWT, ATAT, and context switching.

Keywords: MLFQ, linear regression, scheduling algorithm, dynamic round robin

1) INTRODUCTION:

Data centers, critical to the operations of modern technology infrastructures, face mounting challenges as global data generation increases exponentially. These challenges include resource optimization, energy efficiency, and sustainability. Despite advancements in hardware efficiency and cooling systems, data centers remain significant energy consumers, driving the need for smarter task scheduling and resource allocation techniques [1][2]. The inefficiencies in traditional scheduling algorithms exacerbate computational delays and lead to increased energy consumption, underscoring the need for innovative, dynamic solutions.

In the fields of operating systems and distributed computing, scheduling algorithms play an important role in managing processes and resource allocation that directly influence system performance. Among these, Multilevel Feedback Queues (MLFQ) have emerged as a widely utilized mechanism due to its adaptability and capacity to handle a wide range of tasks and execution times [3]. Traditionally, MLFQ employs static time quanta to regulate task execution within distinct levels, offering a balance between short-task responsiveness and long-task throughput. However, as demands on computational systems grow, there is increasing interest in exploring dynamic scheduling models that can adapt to real-time changes, making use of machine learning and statistical methods to optimize performance.

Recent developments have demonstrated the potential for dynamic approaches, such as those leveraging reinforcement learning, to enhance scheduling efficiency by adapting to workload characteristics in real-time [4][5]. However, these approaches often involve complex, resource-intensive solutions that are difficult to implement in environments where lightweight, low-latency scheduling is paramount, such as in data centers or IoT systems. Furthermore, while round robin scheduling remains a cornerstone of MLFQ configurations, the static nature of its time quanta presents a limitation, especially in highly variable environments [6].

Current research has extensively examined task scheduling across various computational environments, from cloud and fog computing to high-speed data centers. In fog and cloud systems, researchers have proposed novel architectures such as Deep Weighted-Fair Fog Servers (DeepWFFS) and deep reinforcement learning-based solutions that offer dynamic task scheduling [4][7]. These architectures demonstrate the importance of adaptable, granular task management, emphasizing the need for flexibility in handling diverse workloads.

In more traditional settings like data centers, research has focused on optimizing flow completion times and reducing latency for short tasks through innovations in multilevel queue systems [5][6]. These studies highlight the need for scheduling algorithms that can adapt quickly to the network and workload conditions, which current static threshold systems often fail to address. Dynamic scheduling algorithms like D-MLFQ, which autonomously adjust thresholds based on environmental conditions, have shown promise in improving system performance, but gaps remain, particularly in balancing complexity with real-time execution demands.

While the use of deep learning-based dynamic scheduling algorithms has made significant strides, these methods are often resource-intensive and may not be suitable for all environments. Research on lightweight dynamic scheduling algorithms, particularly those that can be easily integrated into existing MLFQ systems, remains limited. Additionally, the role of statistical models like linear regression in dynamically adjusting scheduling parameters, such as time quanta in round robin algorithms, has yet to be explored comprehensively.

Another notable gap is the application of multivariate factors, such as CPU and memory usage, in real-time adjustment of scheduling algorithms. Existing dynamic models, such as D-MLFQ, often focus on single-variable adjustments (e.g., task priority or flow completion time) [5]. However, incorporating multiple system factors could yield a more holistic and adaptive approach to scheduling, potentially reducing overhead while maintaining high efficiency. To better visualize, Figure 1 illustrates how using multiple variables are done from planes to multi-dimensional visualizations. Here, linear models can be as close or as far from the individual axes based on the computed weights and biases.

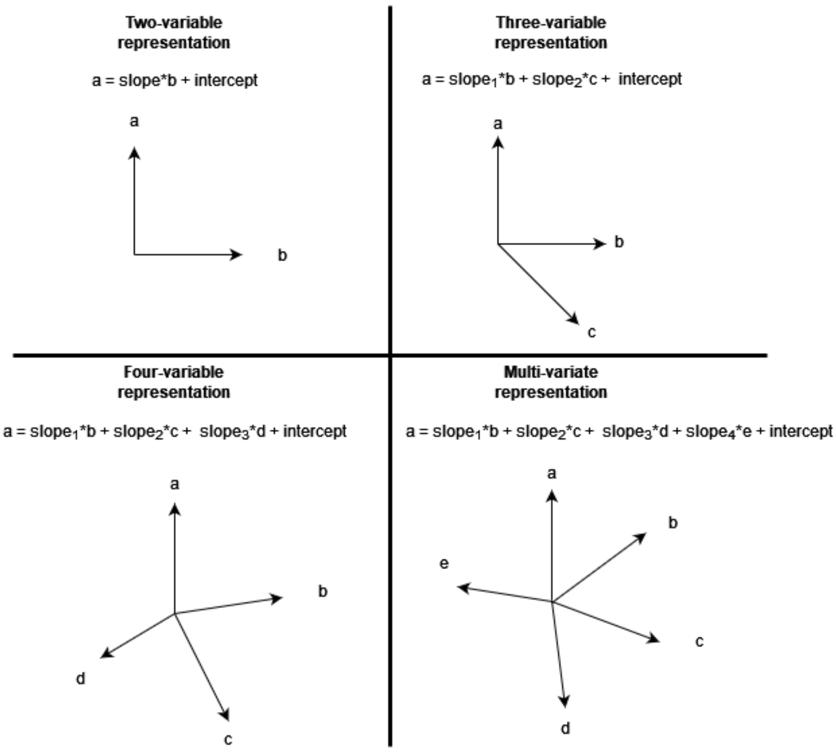

Figure 1 Increment of Axes in Relation to Amount of Variables

Many literature supports the use of dynamic time quanta in round robin algorithms. [8] highlighted that dynamic adjustments in Round Robin scheduling reduce waiting times and improve system responsiveness. They demonstrated how adaptive approaches can outperform static configurations, especially in environments with highly variable workloads where dynamic time quanta can optimize CPU utilization by matching task characteristics to processing resources.[9], for their part, emphasized that resource allocation schemes leveraging dynamic scheduling improve overall system throughput and fairness. Their findings underline the advantages of tailoring scheduling parameters, like time quantum, to workload requirements. [10] stated inefficiencies of static time quantum in Round Robin and proposed improvements using adaptive mechanisms. They noted that dynamic adjustments result in lower metrics.

This research builds on established scheduling theories such as round robin in multilevel feedback queue systems [3], extending them by incorporating regression-based time quantum adjustment. Linear regression models are a natural fit for predicting time quanta based on system metrics, offering a mathematically sound method for dynamic adjustments. Simple linear regression, which predicts based on a single factor, and multivariate linear regression, which uses multiple factors, is employed to explore how different variables influence scheduling efficiency. These concepts support the development of a new scheduling mechanism that adapts to both task and system conditions in real-time.

Linear regression is well-suited for predicting dynamic time quanta due to its simplicity, interpretability, and ability to model relationships between features. [11] demonstrated how regression models can dynamically predict parameters (e.g., CPU burst time) in scheduling algorithms. Their study showed significant improvements in scheduling efficiency when regression models were integrated into scheduling decisions. In the case of [12], the exploration of regression-based models in resource allocation demonstrated their capability to predict dynamic thresholds for scheduling. Their findings show how linear regression models can balance computational efficiency with predictive accuracy. [13] discussed the effectiveness of regression models in optimizing resource allocation systems by predicting parameters based on multivariate inputs. This supports the integration of multivariate regression in adjusting time quantum dynamically. Linear regression is the better fit as the predictions and inputs are mostly numeric data, not categorical.

This study explores an innovative approach to process scheduling by integrating linear regression into the dynamic adjustment of round robin time quanta within MLFQ systems. By utilizing linear regression, the study aims to dynamically calculate time quanta based on real-time factors such as CPU usage, memory usage, network traffic, and task priority, thereby creating a more efficient scheduling mechanism. This study proposes that integrating linear regression could lead to a more efficient scheduling algorithm that dynamically responds to varying task workloads and system conditions, potentially outperforming traditional static approaches.

With the completion of the undertaking, the following are achieved: The identification of system resources that can be modeled using linear regression to predict the dynamic time quantum for round-robin scheduling in a multilevel feedback queue implementation; creation an algorithm using linear regression to optimize system performance in a multilevel feedback queue implementation; and explanation of how the integration of linear regression into round robin dynamic time quanta affect the overall performance of the MLFQ scheduling algorithm.

2) METHODS AND METHODOLOGY:

Following the study framework as illustrated in Figure 2, the study substantiates the application of input (like system resources), processes (MLFQ algorithms with regression models), and output (such as scheduling performance metrics) in the experimental setup. Their structured approaches and emphasis on data-driven processes validate the study's methodology and justifies the relevance of adopting such frameworks for complex systems. [1] highlights how clearly defined inputs, like resource allocation (analogous to system metrics like CPU usage and memory usage), can drive specific outputs such as improved scheduling outcomes (lower ATAT, AWT, and CS). [14] applies IPO logic to model big data integration within the food supply chain. The input is big data from diverse sources, the process involves applying analytics, and the output includes performance improvements in traceability, efficiency, and quality. Meanwhile, [15] proposes a data-driven IPO framework focusing on community collaboration in disaster relief. Inputs include localized data (similar to task-specific system metrics in this study), processes involve data analysis and collaboration, and outputs measure response effectiveness.

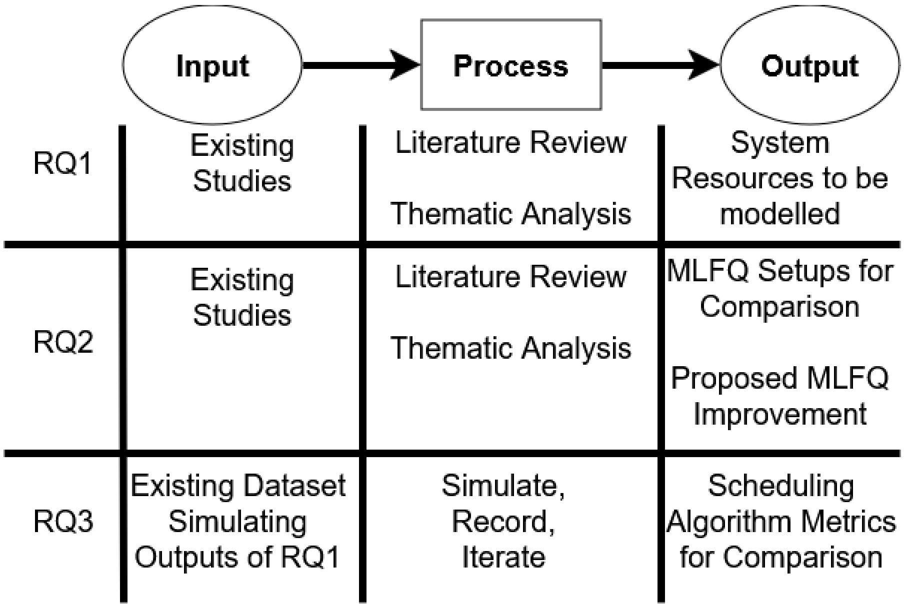

Figure 2 IPO Research Framework

For the first research question, the goal is to identify system resources that can be effectively modeled using linear regression to predict burst times dynamically. The review will focus on metrics such as CPU usage, memory usage, network traffic, power consumption, task type, and task priority. These resources will be evaluated based on their relevance, predictive reliability, and computational efficiency in prior studies related to scheduling optimization. By synthesizing findings from existing literature, the study will provide a comprehensive list of inputs that are well-suited for regression modeling. This approach ensures that the most impactful metrics are selected for inclusion in the experimental design, laying the foundation for the regression-based dynamic scheduling algorithm.

The review of related literature to answer the second research question will examine the methodologies of state-of-the-art studies that employ linear regression or similar predictive models in dynamic scheduling. Special attention will be given to the integration of regression-based dynamic time quantum adjustments in scheduling algorithms like Round Robin in MLFQs. Insights from the literature will guide the conceptualization of a novel algorithmic framework. This algorithm will use linear regression models to dynamically adjust the time quantum for Round Robin layers within MLFQ setups. By leveraging findings from past research, the study will propose a robust algorithm designed to enhance task scheduling efficiency, ensuring its relevance and feasibility before proceeding to experimental validation.

As for the third, the study will employ an experimental setup. The process begins with preparing a dataset that includes system metrics stated in literatures tackled in research question one. This dataset will undergo preprocessing to normalize inputs and will be split into training (70%) and testing (30%) subsets. Regression models will then be trained, including Simple Linear Regression (SLR) models for individual features and a Multivariate Linear Regression (MLR) model for combined metrics. As seen in Figure 3, the experimental phase will involve implementing various MLFQ configurations, with different levels composed of simple algorithms like shortest job first and round robin; and the proposed in research question two. Performance metrics such as Average Waiting Time (AWT), Average Turnaround Time (ATAT), context switches, and Mean Squared Error (MSE) will be collected and analyzed. Statistical tests will be used to compare the static and dynamic setups, identifying the impact of linear regression-based time quantum adjustments on scheduling performance. This experimental validation will provide empirical evidence of the proposed algorithm's effectiveness, linking back to the theoretical foundation established in the previous research questions.

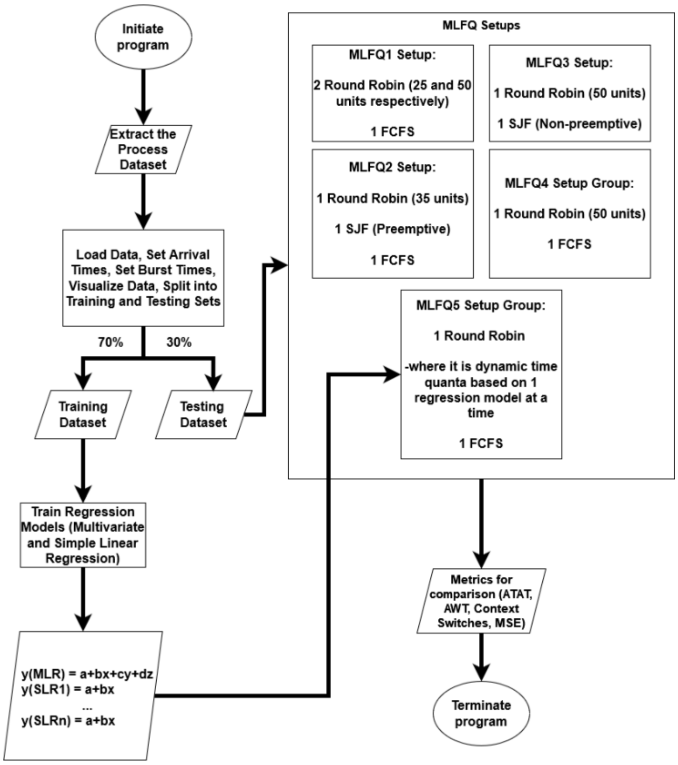

Figure 3 Flowchart of Experimental Setup

The dataset used for this study was retrieved from [19], containing task processes that simulate a high-demand computing environment. The dataset consists of tasks with various burst times, arrival times, and priorities, which were processed by a custom-built simulator created using Google Colab and associated libraries such as pandas, NumPy, and scikit-learn. The simulator applied multiple Multilevel Feedback Queue (MLFQ) configurations, as well as simple and multivariate linear regression (SLR and MLR) models to dynamically adjust the time quantum in specific configurations. The main performance metrics assessed were Average Waiting Time, Average Turnaround Time, and context switching among the MLFQs while MSE was used among the regression models.

3) RESULTS AND DISCUSSIONS:

System Resources that can be Modeled Using Linear Regression to Predict the Time Quantum

CPU usage. This is a critical factor in task scheduling, as it directly reflects the computational demands of tasks within a system. [11] emphasize the role of CPU usage in dynamic scheduling algorithms, demonstrating that predictive models like linear regression can optimize time quanta by accurately modeling workload intensity. Similarly, [12] highlight the significance of CPU utilization in resource allocation, showcasing how CPU usage predictions lead to more efficient scheduling by matching task requirements with available resources. These findings validate the inclusion of CPU usage as a core feature for predicting dynamic time quanta in a multilevel feedback queue system.

Memory usage. This is another essential metric for scheduling algorithms, as it impacts task execution and system responsiveness. [13] identify memory utilization as a key predictor for task performance, noting that linear regression models can effectively capture memory-intensive workload patterns to optimize scheduling decisions. Additionally, [8] discuss the importance of memory availability in adaptive scheduling systems, where dynamic adjustments to time quanta improve overall efficiency. These studies establish memory usage as a valuable input for regression-based time quantum prediction in dynamic scheduling systems.

Network traffic. This plays a pivotal role in task scheduling, particularly for latency-sensitive applications. [10] explore the inclusion of network traffic metrics in scheduling decisions, highlighting how predictive models can address variations in network conditions to ensure fairness and efficiency. Furthermore, [5] underscore the impact of network traffic on scheduling performance in data clusters, advocating for dynamic approaches that leverage real-time traffic predictions. These findings confirm that network traffic is a significant factor for modeling dynamic time quanta in multilevel feedback queue implementations.

Power consumption. This is an increasingly important metric in task scheduling, especially in energy-aware systems. [6] investigate the relationship between power consumption and scheduling performance, showing that predictive modeling can balance energy efficiency and task throughput by adjusting time quanta dynamically. [16] also highlight the role of power metrics in scheduling algorithms, demonstrating how linear regression can optimize resource usage while maintaining system performance. These studies establish power consumption as a vital resource for regression-based dynamic scheduling, enabling systems to achieve both efficiency and sustainability.

Task type. This provides valuable information about workload characteristics, enabling more tailored scheduling decisions. [12] demonstrate the integration of categorical variables like task type into predictive models, showing how these variables can improve scheduling optimization by accounting for workload diversity. [8] further support this by emphasizing the importance of task-specific characteristics in dynamic scheduling systems, where task type helps refine time quantum adjustments to match specific needs. These findings validate the inclusion of task type as a key predictor for regression-based scheduling.

Task priority. This ensures that scheduling algorithms address fairness and responsiveness in multi-task environments. [8] highlight task priority as an essential feature for dynamic scheduling, demonstrating that predictive models can adjust time quanta to meet the urgency of high-priority tasks. Similarly, [12] argue that incorporating priority levels into regression models enhances system adaptability by ensuring that critical tasks receive adequate resources. These studies establish task priority as a crucial factor for modeling dynamic time quantum adjustments in multilevel feedback queue scheduling.

System Resources that can be Modeled Using Linear Regression to Predict the Time Quantum

The proposed algorithm leverages linear regression models to dynamically adjust the time quantum in a Multilevel Feedback Queue (MLFQ) scheduling system, aiming to optimize Average Turnaround Time (ATAT), Average Waiting Time (AWT), context switching, and overall system efficiency.
Based on the literature reviewed, the study will focus on a set of Multilevel Feedback Queue (MLFQ) configurations that are well-supported and relevant to dynamic time quantum adjustments. The most innovative configuration in this study will involve Round Robin with Dynamic Time Quanta, where linear regression models are used to dynamically adjust the time quantum based on real-time system data such as CPU usage, memory load, and task characteristics. Based on insights from [11] and [12], this approach leverages the predictive power of linear regression to adapt the time quantum to the current system state, ensuring that each task receives appropriate CPU time without excessive waiting. This configuration will be tested in the higher-level queues, allowing for dynamic optimization in handling workloads that require efficient processing.

Round Robin is a widely used algorithm where each task is assigned a fixed time quantum. Tasks are processed in circular order until they completed or are requeued if they have not finished within their time slice. Round Robin is known for its fairness, as each task receives equal CPU time. The use of fixed time quanta simplifies the scheduling process, making it ideal for static workloads. It is typically combined with Shortest Job First (SJF) or First-Come-First-Served (FCFS) in MLFQ configurations.

For other queue setups for comparison, the study will implement Round Robin (RR) with fixed or dynamic Time Quanta. To support [17] and [6], traditional Round Robin is widely used due to its fairness in distributing CPU time equally across tasks. However, the fixed nature of the time quantum can lead to inefficiencies, particularly when tasks have widely varying burst times. Round Robin will be tested in combination with higher-level algorithms like Preemptive SJF, creating a balanced system where time-sensitive tasks are handled at the top levels, and general tasks receive fair time distribution in lower-priority queues.

The levels might employ Preemptive Shortest Job First (SJF). This approach, as highlighted by [10] and [9], is effective in reducing response times by allowing shorter tasks to preempt longer ones, thus improving overall system throughput and responsiveness. Preemptive SJF is essential for tasks that require quick turnaround times and is typically used in the upper levels of the MLFQ system to handle high-priority processes.

First-Come-First-Served is a simple, non-preemptive algorithm where tasks are processed in the order of arrival, without consideration of their priority or burst time. FCFS is straightforward and easy to implement, making it suitable for lower-level queues where tasks are not time-sensitive. However, it suffers from the "convoy effect," where longer tasks delay shorter ones. This is typically combined with Round Robin or SJF at higher levels. FCFS is used for the lowest-priority tasks that can tolerate longer waiting times. [10] and Chandiramani et al.'s (2019) analysis of MLFQ systems highlights FCFS as the default choice for the lowest priority queue. This is because FCFS ensures that tasks at the lowest level, which often have the least urgency, are processed without interruptions, reducing the complexity of scheduling at that layer.

In summary, the MLFQ configurations exhibited in this study will include Preemptive SJF, Non-preemptive SJF, Round Robin with Fixed Time Quanta, and the novel Round Robin with Dynamic Time Quanta powered by linear regression models. These configurations cover a range of scheduling strategies that will allow the study to thoroughly evaluate the effectiveness of dynamic time quantum adjustments in improving scheduling performance.

In the first setup (MLFQ1), the system employed two rounds of Round Robin with fixed time quanta, followed by First-Come-First-Served (FCFS). The time quanta of the round robin levels are 25 and 50 respectively. This setup is most common in examples and literature of a depiction on MLFQs [10][18]. In the second setup the system applied one Round Robin cycle with a time quantum of 50, followed by Preemptive Shortest Job First (SJF), and finally FCFS. This is synthesized from the readings stated above. In the third setup, Non-Preemptive SJF replaced the preemptive version, alongside one Round Robin cycle prior while the round robin time quanta is 25. The fourth is a static version of the proposed fifth setup where it would be one (1) level of round robin with a time quantum of 50 and the last level being FCFS. The fifth setup (and those proceeding) introduced dynamic time quantum adjustments powered by a linear regression model. This configuration applied one Round Robin cycle with time quantum dynamically adjusted based on the system's state and task characteristics, followed by FCFS for longer tasks. Figure 3 is a visualization of these.

Performance of the MLFQ Setups

The findings indicate that integrating regression-based dynamic time quantum adjustments into MLFQ scheduling can significantly enhance performance. Among the traditional methods, MLFQ Block 2 (Round Robin, Preemptive SJF, and FCFS) was the most effective, but it was still outperformed by all configurations of MLFQ Block 5. This suggests that regression models, particularly when based on CPU usage, offer substantial improvements in scheduling efficiency over static configurations. To further cement the implications that dynamic round robin through linear regression in MLFQs deliver better results, nine more test cases were created. Through incrementing the random_state variable in the split function of the simulator by ten (10) in the splitting of the training and test datasets, multiple cases are created and their metrics collected after running through the simulator. As seen in Figure 4 and Figure 5, the ATAT and AWT are visualized per setup per test case.

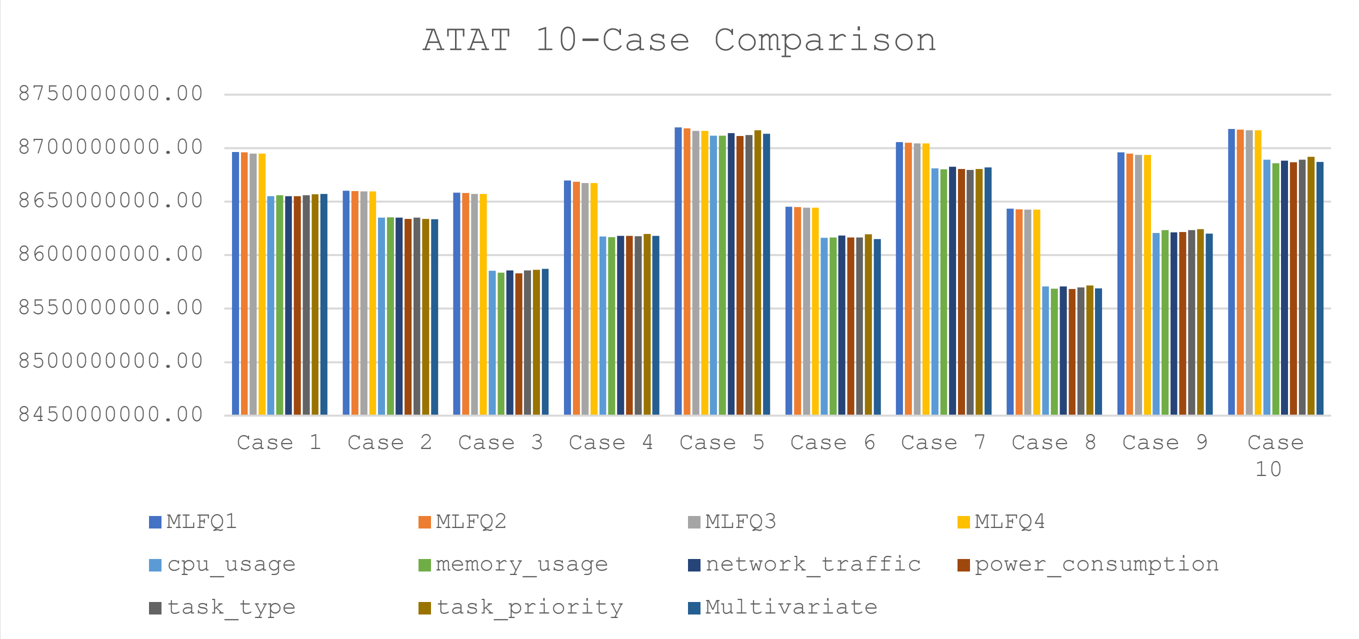

Figure 4 ATAT Among the Ten (10) Test Cases

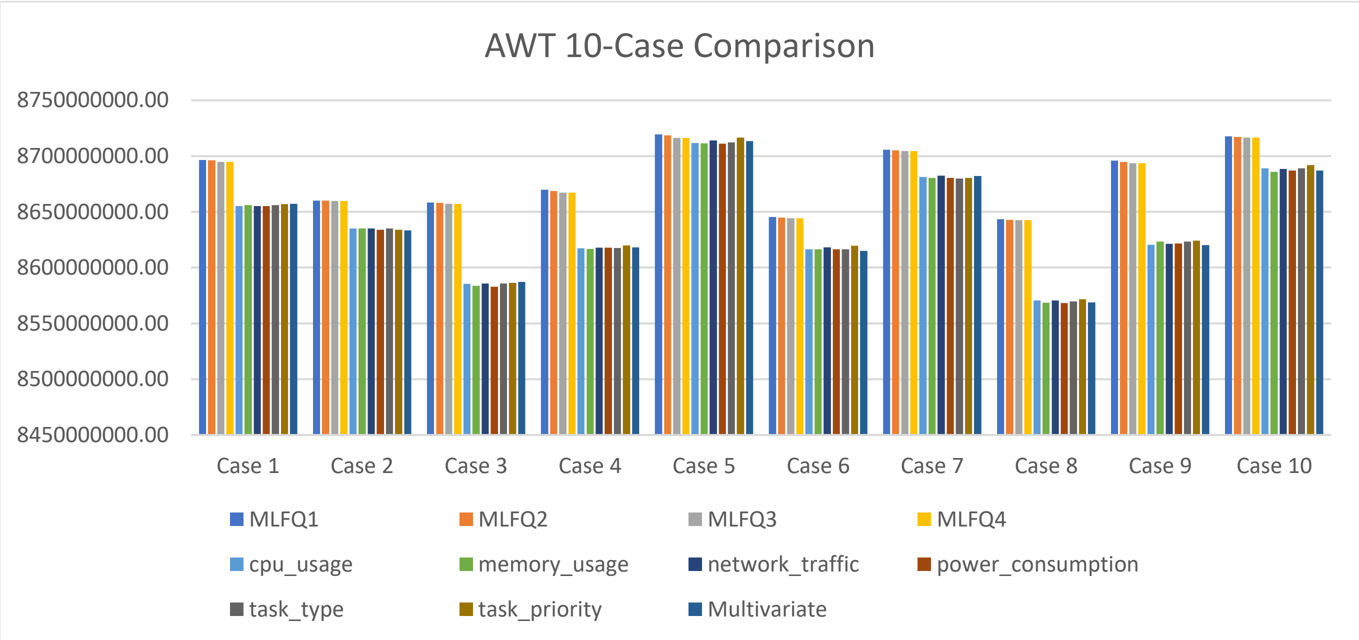

Figure 5 AWT Among the Ten (10) Test Cases

The means for MLFQ1, MLFQ2, MLFQ3, and MLFQ4 are very close, all approximately 8.68×10^9 ms, showing minimal variation between static configurations. The mean ATAT for the regression-based models ranges from 8.638×10^9 ms to 8.640×10^9 ms, slightly lower than the traditional MLFQ configurations, indicating better turnaround time due to dynamic time quantum adjustments. Overall, the Shapiro-Wilk Test proves that the data are normal.

These reveal that the regression-based dynamic configurations consistently achieve lower mean values compared to static MLFQ configurations, indicating improved scheduling efficiency. Regression models leveraging features like cpu_usage and memory_usage demonstrate the lowest variability (standard deviations around 4.56×10^7 ms) and thus exhibit more consistent performance. The Shapiro-Wilk test confirms normality for all configurations, with p-values exceeding 0.05, making the data suitable for parametric comparisons.

The Greenhouse-Geisser Correction is applied because the assumption of sphericity (equal variances of differences) is violated. For both AWT and ATAT, the p-value is less than .001, which is highly significant. This result indicates that there are statistically significant differences in AWT and ATAT across the MLFQ configurations.

Performing repeated measures ANOVA on separate AWT and ATAT datasets, the results in Table 1 are created. Since the p-values for both AWT and ATAT are < .001, it can be concluded that the different MLFQ configurations have a significant impact on both Average Waiting Time and Average Turnaround Time. In practical terms, this means that the choice of MLFQ configuration leads to meaningful differences in scheduling performance. To have a closer look at the differences, post hoc tests were done to the ATAT and AWT results. The post hoc test results for ATAT and AWT indicate significant differences between the traditional MLFQ configurations (MLFQ1, MLFQ2, MLFQ3, and MLFQ4) and all regression-based MLFQ5 models, with p-values less than 0.001. The differences are consistently large showing that the regression-based dynamic time quantum adjustments significantly reduce turnaround times compared to static configurations. However, among the MLFQ5 configurations, no significant differences are observed, as all pairwise comparisons yield non-significant p-values (p > 0.05). This suggests that while the dynamic models outperform static ones, the specific feature or regression model used does not drastically affect performance. This reinforces the effectiveness of dynamic adjustments as a group while indicating a relative equivalence between different regression-based approaches. For details, see appendices C and D for the results of post hoc tests.

Table 1. ANOVA Results

Cases	Sphericity Correction	Sum of Squares	df	Mean Square	F	p
ATAT	Greenhouse-Geisser	$4.456\times10^{+16}$	1.011	$4.408\times10^{+16}$	29.745	< .001
Residuals (ATAT)	Greenhouse-Geisser	$1.348\times10^{+16}$	9.099	$1.482\times10^{+15}$		
AWT	Greenhouse-Geisser	$4.456\times10^{+16}$	1.011	$4.408\times10^{+16}$	29.745	< .001
Residuals (AWT)	Greenhouse-Geisser	$1.348\times10^{+16}$	9.099	$1.482\times10^{+15}$		

Among the static configurations, MLFQ1 demonstrates the highest average number of context switches at 74,394, with a standard deviation of 168.18. This reflects the inefficiency of using two static Round Robin layers, which cause frequent task preemptions and significant scheduling overhead. MLFQ2, on the other hand, performs better with an average of 58,055 context switches and a lower standard deviation of 133.29. The inclusion of Preemptive Shortest Job First (SJF) in MLFQ2 reduces unnecessary switches by prioritizing shorter tasks, resulting in more efficient scheduling compared to MLFQ1.

MLFQ3 and MLFQ4 exhibit the lowest average context switches among the static configurations, both at 44,642. Their improved performance stems from the use of only one Round Robin layer in combination with Non-preemptive SJF (MLFQ3) and a static Round Robin layer (MLFQ4). This minimizes the overhead associated with frequent preemptions and ensures a balance between task prioritization and system fairness. Furthermore, their lower standard deviations (~75.95) indicate greater consistency in reducing context switches compared to MLFQ1 and MLFQ2.

As shown in Table 2, the regression-based MLFQ5 configurations show comparable performance to MLFQ3 and MLFQ4, with averages ranging between 44,641 and 44,646 context switches. These setups utilize dynamic time quantum adjustments based on real-time feature predictions, which effectively maintain efficiency while introducing adaptability. However, slightly higher variability (standard deviations between 95.88 and 97.97) reflects the dynamic nature of these adjustments.

Table 2. Amount of Context Switches

Setup	MLFQ 1	MLFQ 2	MLFQ 3	MLFQ 4	cpu_usage	memory_usage	network_traffic	power_consumption	task_type	task_priority	Multivariate
Case 1	74167	57915	44572	44572	44545	44546	44548	44551	44548	44545	44551
Case 2	74420	57982	44634	44634	44644	44634	44634	44637	44632	44624	44625
Case 3	74534	58172	44689	44689	44708	44709	44707	44707	44707	44709	44713
Case 4	74416	58004	44644	44644	44654	44649	44659	44657	44651	44649	44646
Case 5	74420	58021	44634	44634	44630	44629	44632	44626	44628	44625	44635
Case 6	74341	58000	44690	44690	44686	44689	44685	44695	44689	44698	44700
Case 7	74262	58061	44554	44554	44547	44542	44543	44555	44544	44549	44557
Case 8	74203	57904	44529	44529	44493	44491	44489	44501	44494	44485	44493
Case 9	74435	58142	44695	44695	44704	44700	44708	44708	44700	44703	44704
Case 10	74745	58345	44780	44780	44820	44820	44821	44824	44820	44820	44826
Average	74,394	58,055	44,642	44,642	44,643	44,641	44,643	44,646	44,641	44,641	44,645
SD	168.1798	133.2935	75.94655	75.94655	95.87892	96.6281	97.16332	94.62376	95.56737	97.96944	96.65287

In the case of MSEs between the linear regression models, little differences can be found. The uniform performance of the regression models underscores the versatility of the features in predicting burst times for dynamic scheduling. While individual features like cpu_usage and task_priority perform slightly better in terms of consistency, the multivariate model offers a comprehensive perspective by incorporating multiple system metrics. These findings support the feasibility and reliability of regression-based dynamic time quantum adjustments in scheduling systems. It does however mean that there is an unnecessary number of computations should multivariate be used in modeling such predictive functions. Thus, simple linear regression should be used in practical implementations.

4) CONCLUSION:

The variables that can be modelled in linear regression are CPU usage, memory usage, network traffic, power consumption, task type, and task priority. These are effective predictors for dynamic time quantum adjustments in round-robin scheduling. The proposed algorithm composes of Round Robin with dynamic time quanta using linear regression and First Come, First Served implemented in Multilevel Feedback Queue systems. The regression-integrated MLFQ significantly outperforms traditional approaches in minimizing ATAT, AWT, and context switching, thereby improving overall system efficiency.

5) ACKNOWLEDGEMENT:

We thank Luis William C. Meing and Dionisio R. Tandingan Jr. for their contributions to this work. Special thanks to the University of the Cordilleras for their assistance in the completion of these works.

REFERENCES:

[1] Deng, J., Huang, S., Wang, L., Deng, W., & Yang, T. (2022). Conceptual framework for smart health: A multi-dimensional model using IPO logic to link drivers and outcomes. International Journal of Environmental Research and Public Health, 19(24), 16742. https://doi.org/10.3390/ijerph192416742

[2] Tripathy, S. S., Mishra, K., Roy, D. S., Yadav, K., Alferaidi, A., Viriyasitavat, W., Sharmila, J., Dhiman, G., & Barik, R. K. (2023). State-of-the-Art load Balancing Algorithms for Mist-Fog-Cloud Assisted Paradigm: A review and future directions. Archives of Computational Methods in Engineering, 30(4), 2725–2760. https://doi.org/10.1007/s11831-023-09885-1

[3] Pemasinghe, S., & Rajapaksha, S. (2022). Comparison of CPU scheduling algorithms: FCFS, SJF, SRTF, round robin, priority based, and multilevel queuing. 2022 IEEE 10th Region 10 Humanitarian Technology Conference (R10-HTC), 4, 318–323. https://doi.org/10.1109/r10-htc54060.2022.9929533

[4] Heidarpour, A. R., Heidarpour, M. R., Ardakani, M., Tellambura, C., & Uysal, M. (2024). DeepWFFS: Enhancing FOG computing efficiency through Multi-Queue architecture and intelligent Controller for task

[5] Wang, Z., Yan, K., Chang, G., Huang, C., Xiao, S., & Yang, Y. (2023). Towards adaptive adjusting and efficient scheduling coflows based on deep reinforcement learning. 2023 IEEE 29th International Conference on Parallel and Distributed Systems (ICPADS). https://doi.org/10.1109/icpads60453.2023.00278

[6] Iqbal, M. S., & Chen, C. (2023). AMFQ: Approximating multi-level feedback queue scheduling at line rate for data center networks. GLOBECOM 2023 - 2023 IEEE Global Communications Conference. https://doi.org/10.1109/globecom54140.2023.10437800

[7] Birman, Y., Ido, Z., Katz, G., & Shabtai, A. (2021). Hierarchical deep reinforcement learning approach for Multi-Objective scheduling with varying queue sizes. 2021 International Joint Conference on Neural Networks (IJCNN). https://doi.org/10.1109/ijcnn52387.2021.9534433

[8] Barnawi, A., Sakr, S., Xiao, W., & Al-Barakati, A. (2020). The views, measurements and challenges of elasticity in the cloud: A review. Computer Communications, 154, 111–117. https://doi.org/10.1016/j.comcom.2020.02.010

[9] Qureshi, M. S., Qureshi, M. B., Fayaz, M., Mashwani, W. K., Belhaouari, S. B., Hassan, S., & Shah, A. (2020). A comparative analysis of resource allocation schemes for real-time services in high-performance computing systems. International Journal of Distributed Sensor Networks, 16(8), 155014772093275. https://doi.org/10.1177/1550147720932750

[10] Jodayree, M., Abaza, M., & Tan, Q. (2019). A predictive workload balancing algorithm in cloud services. Procedia Computer Science, 159, 902–912. https://doi.org/10.1016/j.procs.2019.09.250

[11] Yadav, M., & Mishra, A. (2023). An enhanced ordinal optimization with lower scheduling overhead based novel approach for task scheduling in cloud computing environment. Journal of Cloud Computing Advances Systems and Applications, 12(1). https://doi.org/10.1186/s13677-023-00392-z

[12] Rocha, H. R., Honorato, I. H., Fiorotti, R., Celeste, W. C., Silvestre, L. J., & Silva, J. A. (2020). An artificial intelligence based scheduling algorithm for demand-side energy management in smart homes. Applied Energy, 282, 116145. https://doi.org/10.1016/j.apenergy.2020.116145

[13] Huang, H. Y., Kim, K. T., & Youn, H. Y. (2020). Determining node duty cycle using Q-learning and linear regression for WSN. Frontiers of Computer Science, 15(1). https://doi.org/10.1007/s11704-020-9153-6

[14] Margaritis, I., Madas, M., & Vlachopoulou, M. (2022). Big data applications in food supply chain management: A conceptual framework. Sustainability, 14(7), 4035. https://doi.org/10.3390/su14074035

[15] Aldowaish, A., Kokuryo, J., Almazyad, O., & Goi, H. C. (2022). Environmental, Social, and Governance Integration into the Business Model: Literature Review and Research Agenda. Sustainability, 14(5), 2959. https://doi.org/10.3390/su14052959

[16] Khadilkar, A., Kasodekar, K. S., Sharma, P., & Priyadarshini, J. (2018). Intelligent traffic light scheduling using linear regression. In Advances in intelligent systems and computing (pp. 329–335). https://doi.org/10.1007/978-981-13-1822-1_30

[17] Mulder, D., Ssempala, J., Walton, T., Parker, B., Brough, S., Bush, S., Boroojerdi, S., & Tang, J. (2020). Impact of quantum values on multilevel feedback queue for CPU scheduling. 2020 Intermountain Engineering, Technology and Computing (IETC), 1–4. https://doi.org/10.1109/ietc47856.2020.9249188

[18] Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating system concepts (10th ed.). Wiley.

[19] Cloud Computing Performance Metrics. (2023). [Dataset]. In A. Khan, Kaggle. https://www.kaggle.com/datasets/abdurraziq01/cloud-computing-performance-metrics