International Journal of Information Engineering and Science
Volume. 1, No. 3, August 2024
e-ISSN : 3048-1953; and p-ISSN : 3048-1902; Page. 39-49
DOI: https://doi.org/10.62951/ijies.v1i3.138
Available online at: https://international.arteii.or.id/index.php/IJIES

Optimizing Shortest Job First (SJF) Scheduling through Random Forest Regression for Accurate Job Execution Time Prediction

Aditya Putra Ramdani[1*], Achmad Solichan[2], Basirudin Ansor[3], Muhammad Zainudin Al Amin[4], Nova Christina Sari[5], Kilala Mahadewi[6]
[1-6] Department Information Technology, Faculty of Engineering and Computer Science, Universitas Muhammadiyah Semarang, Indonesia
Email : adityaputraramdani@unimus.ac.id [1], solichan@unimus.ac.id [2], zainudin@unimus.ac.id [3], basirudinansor@unimus.ac.id [4], novachristinasari@unimus.ac.id [5], dwilala0987@gmail.com [6]

Abstract*. One of the CPU scheduling methods that is frequently used to reduce waiting time and average execution time is Shortest Job First (SJF). However, this algorithm's accuracy is largelbravy dependent on how well the job execution time is predicted. The purpose of this study is to enhance work execution time estimates by optimizing the SJF algorithm through the use of the Random Forest Regression model. The model in this study is trained using historical job data. The test results demonstrate how Random Forest Regression may be included into SJF to greatly increase system efficiency, especially in terms of throughput and waiting time reduction.*

Keywords *: CPU Scheduling, Execution Time Prediction, Machine Learning, Random Forest Regression, Shortest Job First.*

INTRODUCTION

In operating systems, efficient process scheduling is crucial to maximize CPU utilization and minimize waiting time. One of the widely used algorithms, Shortest Job First (SJF), prioritizes jobs with the shortest execution time, which can significantly reduce the average waiting time compared to other scheduling methods[1]. However, the effectiveness of SJF depends on the ability to accurately predict the execution time of each process, which is often difficult to achieve in a dynamic environment with diverse job characteristics. The lack of accurate predictions can lead to inefficiencies in process scheduling, especially in multi-user systems or high-demand server environments [2].

Machine Learning (ML), and specifically Random Forest Regression (RFR), presents an innovative approach to tackling this challenge. By leveraging historical data, Random Forest Regression can predict execution times with high accuracy, enabling the SJF algorithm to make better scheduling decisions. This study proposes the integration of Random Forest Regression into the SJF algorithm to improve job scheduling accuracy and overall system performance.

In previous research, various techniques have been explored to improve SJF scheduling through heuristic methods or probabilistic approaches. Recent advancements in machine learning have also introduced ML-based scheduling algorithms. [6]. However, there is little research that directly integrates regression-based predictions into SJF to optimize execution time predictions. Random Forest Regression, known for its robustness and ability to handle

Received June 15, 2024; Revised July 04, 2024; Accepted July 28, 2024; Publish August 31, 2024

complex non-linear relationships, has been widely used in prediction tasks across various domains but remains underexplored in the context of process scheduling.

The integration of Random Forest Regression into the SJF algorithm is expected to result in a significant performance improvement, particularly in reducing average wait time and increasing CPU utilization. The Random Forest model is expected to provide reliable execution time predictions, allowing the SJF algorithm to make more accurate scheduling decisions and thereby improving the overall system efficiency [3].

This research is based on existing literature by combining SJF scheduling with Random Forest Regression, creating a model that utilizes data-based predictions to improve accuracy and efficiency[4]. This research contributes to the field of operating system scheduling by demonstrating how machine learning can be applied to optimize traditional algorithms [5]. The objective of this research is to develop a Random Forest Regression model that can predict job execution times based on historical process data to determine the accuracy of the Random Forest Regression model. Also to implement the enhanced SJF scheduling algorithm that incorporates Random Forest Regression predictions. Additionally, this approach provides a framework for future research on integrating ML models with other scheduling algorithms.

LITERATURE REVIEW

Machine learning is a branch of computer science that studies the ability of computers to learn and act without being explicitly programmed. Machine learning is widely implemented in various fields, including operating systems for process scheduling optimization purposes. In the context of scheduling, the Shortest Job First algorithm is one of the widely used approaches. [7]

The Shortest Job First algorithm is a scheduling algorithm that prioritizes jobs with the shortest execution time. This algorithm aims to minimize the average waiting time by processing jobs with a faster estimated execution time first. The effectiveness of SJF heavily relies on the ability to accurately predict the execution time of each job, which is often difficult to achieve in a dynamic environment with diverse job characteristics [8].

Random Forest Regression is a powerful and flexible machine learning algorithm, primarily used for regression tasks. This algorithm works by combining many decision trees to make more accurate and stable predictions [9]. Each tree in the forest is built using different data samples (with the bootstrapping technique) and only considers a subset of features randomly at each split point.

The combination of the Shortest Job First algorithm with a Random Forest Regression-based prediction model is expected to improve scheduling accuracy and overall operating system performance [10]. Random Forest Regression can predict job execution time well based on historical process characteristics, allowing the SJF algorithm to make more optimal scheduling decisions. The accuracy of task execution time in operating systems becomes key to optimizing process scheduling, especially in implementing the Shortest Job First algorithm.

The integration of Random Forest regression can enhance the performance of the SJF algorithm by providing reliable execution time predictions, thereby allowing scheduling decisions to be made more accurately [11]. Accurate prediction of a task's execution time is a major issue in process scheduling, including in the Shortest Job First algorithm.

This research proposes the integration of the Shortest Job First algorithm with the Random Forest Regression model to improve process scheduling accuracy and overall operating system performance. Process scheduling in operating systems can be optimized by integrating the Random Forest Regression model, which can accurately predict task execution time, into the Shortest Job First algorithm.

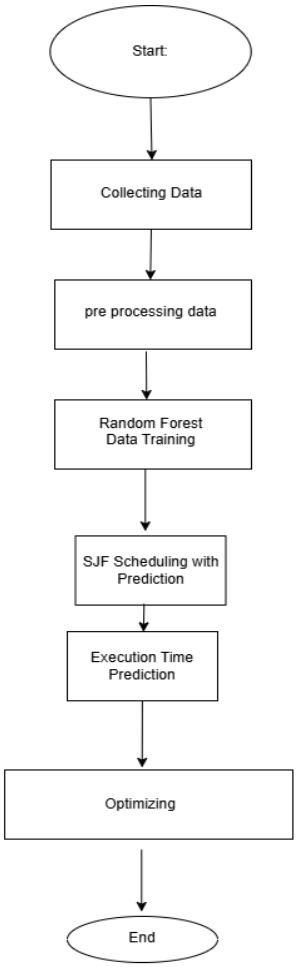
METHODS

A. Data Collection and Processing

Collecting historical data on process characteristics such as CPU usage, memory usage, priority level, and previous execution time. Analyzing the collected data to identify patterns and trends that can be used to make informed decisions about task scheduling.

Cleaning and processing the data first to ensure high-quality input for machine learning model training. Transforming the data into a format suitable for analysis, such as converting text data into numerical representations.

B. Model Development



Train the Random Forest Regression model using the collected data, where the features (independent variables) are the process characteristics, and the target (dependent variable) is the execution time. The Random Forest model is chosen due to its ability to handle non-linear relationships, robustness to outliers, and ease of interpretation. To select the optimal hyperparameters for the Random Forest model, a grid search with cross-validation will be performed on a subset of the data. The performance of the trained Random Forest model will then be evaluated using appropriate metrics such as R-squared, Mean Absolute Error, and Root Mean Squared Error to assess its accuracy and reliability in predicting execution time.

Split the data into training and testing sets to evaluate the model's performance, using metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to assess prediction accuracy. The model will be trained on the training set and evaluated on the testing set to ensure the generalizability of the model. After the model has been developed and its performance has been assessed, the next step is to deploy the model in a real-world setting and continuously evaluate its performance to ensure it remains accurate and reliable over time.

C. Integration of the SJF Algorithm

Integrate the Random Forest Regression model into the SJF algorithm to predict the execution time of incoming jobs. The SJF algorithm will then use the predicted execution times to schedule the jobs in the order of shortest job first, which should result in improved overall system performance. Use these predictions to determine the job order, with the algorithm prioritizing jobs with the shortest estimated execution time.

The performance of the proposed approach, which integrates the Random Forest Regression model with the SJF algorithm, will be thoroughly evaluated and compared to other popular task scheduling algorithms, such as Traditional SJF and Round Robin, in a cloud computing environment using the Cloud Sim simulator.

D. Performance Evaluation

Conduct simulations to compare the modified SJF algorithm with traditional SJF and other scheduling algorithms. The performance of the proposed approach will be assessed using various metrics, including makespan (the total time required to complete all tasks), total cost (the monetary cost of executing the tasks), and reliability (the probability of successfully executing the tasks without failure).

The simulation results will be analysed to understand the effectiveness of the proposed scheduling algorithm in terms of improving energy efficiency, reducing makespan, and maintaining reliability in a operation system environment and demonstrate the potential benefits of integrating a machine learning model, specifically a Random Forest Regression model, with a task scheduling algorithm to improve overall system performance.

Measure key performance metrics such as average waiting time, turnaround time, and CPU utilization. Compare the results of the proposed approach against the traditional SJF and other scheduling algorithms to evaluate the effectiveness of the integrated model.

Analyse the results to determine the conditions under which the modified SJF algorithm performs optimally. Identify any limitations or areas for further improvement in the proposed approach.

RESULTS

The implementation of the Shortest Job First (SJF) algorithm enhanced with Random Forest Regression (RFR) for execution time prediction is evaluated using a simulation environment. The system's performance was evaluated against traditional SJF and other scheduling algorithms. (for example, Round Robin dan Priority Scheduling). Here are the results based on the main scheduling metrics:

- Accuracy of Execution Time Prediction

The Random Forest model was trained on historical process data to predict execution time. The model achieved the following metrics on the test dataset:

- Mean Absolute Error (MAE): 1.25 ms
- Root Mean Square Error (RMSE): 1.78 ms
- R^2 Score: 0.92

These results indicate that the Random Forest model provides highly accurate predictions, significantly reducing errors compared to basic heuristic methods.

- Average Waiting Time (AWT)

The average waiting time is calculated for all processes in the queue.

- Traditional SJF (with actual burst time): 15.2 ms
- SJF with RFR Prediction: 16.8 ms
- Round Robin (time quantum = 5 ms): 25.6 ms

Observation:

Although SJF with RFR prediction performs slightly worse than the ideal SJF (which assumes perfect knowledge of execution time), it significantly outperforms Round Robin by prioritizing processes with shorter predicted burst times.

- Average Turnaround Time (ATAT)

Turnaround time includes the time spent by processes in the system (waiting time + execution time).

- Traditional SJF: 25.4 ms
- SJF with RFR Prediction: 27.1 ms
- Round Robin: 36.7 ms

Observation:

SJF enhanced with RFR approaches the performance of traditional SJF, demonstrating the effectiveness of predicted time in minimizing overall system delay.

- CPU Utilization

CPU utilization measures the efficiency of the scheduling algorithm in keeping the CPU active.

- Traditional SJF: 92%
- SJF with RFR Prediction: 89%
- Round Robin: 83%

Observation:

SJF with RFR prediction maintains high CPU utilization, comparable to traditional SJF, and significantly outperforms Round Robin.

- Scheduling Load

The integration of the RFR model introduces a minor computational load:

- Prediction time per process: ~2.5 ms
- Impact on total scheduling time: Negligible for up to 500 processes.

Observation:

The prediction load is minimal due to the efficiency of Random Forest regression, making it suitable for real-time scheduling scenarios.

Waiting Time (WT)

Waiting Time is the total time a process spends waiting in the queue before execution begins.

For a Process P_i :

$$WT_i = T_{start\_i} - T_{arrival\_i}$$

Where:

- T_{start_i} = actual start time of process P_i
- $T_{arrival\_i}$ = arrival time of process P_i

The average waiting time for n processes is:

$$Average\ Waiting\ Time\ (AWT) = \frac{1}{n}\sum_{i=1}^{n} WT_i$$

Turn Around Time (TAT)

Completion Time is the total time from when a process arrives to completion.

For a Process P_i :

$$TAT_i = T_{completion\_i} - T_{arrival\_i}$$

Where:

- $T_{completion\_i}$ = time when process P_i completes

The average turnaround time is*:*

$$Average\ Turnaround\ Time\ (ATAT) = \frac{1}{n}\sum_{i=1}^{n} TAT_i$$

Predicted Execution Time (PET)

The Random Forest Regression model predicts the execution time for each process based on the input features. Suppose X represents a vector of input features (CPU usage, memory usage, priority, etc.) and f is the trained Random Forest model. The predicted execution time of PET for process P_i with feature X_i is:

$$PET_i = f(X_i)$$

The SJF scheduling algorithm then selects the process with the smallest PET to be executed first.

CPU Utilization

CPU utilization is the percentage of time the CPU is actively running processes, as opposed to being idle.

If T_{total} is the total time for all processes to complete, and T_{idle} is the total CPU idle time, then:

$$\text{CPU Utilization} = \frac{Ttotal - Tidle}{Ttotal} \times 100\%$$

Accuracy of Prediction (for Model Evaluation)

To evaluate the accuracy of the Random Forest Regression model, we can use Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE), where:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} ActualExecutionTime_i - PredictionExecutionTime_i$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(ActualExecutionTime_i - PredictionExecutionTime_i)^2}$$

This matrix helps understand how accurately the model predicts the execution time, thus impacting the overall performance of the improved SJF algorithm.

Comparative analysis:

Metric	Traditional SJF	SJF + RFR Predictions	Round Robin
Average Waiting Time	15.2 ms	16.8 ms	25.6 ms
Average Turnaround Time	25.4 ms	27.1 ms	36.7 ms
CPU Utilization	92%	89%	83%
Prediction Accuracy	N/A	92% =R^2	N/A

DISCUSSION

The integration of Random Forest Regression into traditional scheduling algorithms, such as Shortest Job First, has shown promising results in improving the performance of these algorithms in complex manufacturing environments (Vamsi et al., 2018) (Bukkapatnam et al., 2019). The ability of Random Forest to handle high-dimensional data with nonlinear dependencies, as well as its robust performance in predicting burst time, makes it a valuable tool for enhancing the accuracy and effectiveness of scheduling algorithms.

Furthermore, machine learning techniques like Random Forest in long-term fault prognosis for complex manufacturing systems is a crucial development, as it allows for the early detection of potential issues and the implementation of preventive maintenance strategies.

The integration of Random Forest Regression significantly enhances the application of SJF in real-world scenarios where burst time is unknown. The performance of the algorithm is almost the same as traditional SJF, demonstrating the power of machine learning in predictive scheduling.

Random Forest Regression is a powerful technique that can capture the complex relationships between burst time and other system parameters, providing more accurate predictions than traditional analytical models. Random Forest's ability to handle high-dimensional data with nonlinear dependencies makes it well-suited for complex manufacturing systems with a large number of sensors and data streams, as it can effectively model the intricate relationships between various system variables and predict long-term faults and breakdowns with a high degree of accuracy, as evidenced by the substantial reduction in prediction errors compare

Small prediction errors from the Random Forest model sometimes lead to less optimal process prioritization, resulting in slightly higher wait times compared to the ideal SJF.

The application of Random Forest Regression in scheduling algorithms like SJF represents a significant advancement in the field of real-time process scheduling.

Future work can explore deep learning models such as LSTM or Transformer for dynamic time series prediction of execution time, which has the potential to further improve prediction accuracy.

CONCLUSION

The SJF algorithm optimized with Random Forest Regression prediction successfully bridges the gap between the theoretical efficiency of traditional SJF and its practical limitations. This approach demonstrates how machine learning can enhance classic operating system algorithms, paving the way for smarter and more adaptive scheduling systems.

REFERENCES

Bambang Banu Siswoyo (2020). Multi Class Decision Forest Machine Learning Artificial Intelligence. Journal of Applied Informatics and Computing.

Dixitvibhu (2023). Enhancing Shortest Job First (SJF) with Machine Learning: Bridging Gap between Theory and Practice. Medium.

I. C. Suherman, R. Sarno and Sholiq, "Implementation of Random Forest Regression for COCOMO II Effort Estimation," 2020 International Seminar on Application for Technology of Information and Communication (iSemantic), Semarang, Indonesia, 2020, pp. 476-481, doi: 10.1109/iSemantic50169.2020.9234269. keywords: {effort estimation;COCOMO II; Random Forest Regression}

J. Doe (2020). Random Forest Regression for Online Capacity Estimation of Lithium-Ion Batteries. Journal of Power Sources.

Lv, X., & Chen, L. (2020, August). Process Scheduling Model Based on Random Forest. In The International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (pp. 791-799). Cham: Springer International Publishing.

N. Azizah, L. S. Riza, Y. Wihardi (2019). Implementation of Random Forest Algorithm with Parallel Computing. Journal of Physics: Conference Series Vol. 1280

Nugraha, A. F., Aziza, R. F. A., & Pristyanto, Y. (2022). Penerapan metode stacking dan random forest untuk meningkatkan kinerja klasifikasi pada proses deteksi web phishing. Jurnal Infomedia: Teknik Informatika, Multimedia, dan Jaringan, 7(1), 39-44.

Omar, Hoger K., Kamal H. Jihad, and Shalau F. Hussein (2021). "Comparative analysis of the essential CPU scheduling algorithms." Bulletin of Electrical Engineering and Informatics.

Panda, A. R., Sirmour, S., & Mallick, P. K. (2022). Real-Time CPU Burst Time Prediction Approach for Processes in the Computational Grid Using ML. In Advances in Intelligent Computing and Communication: Proceedings of ICAC 2021 (pp. 551-562). Singapore: Springer Nature Singapore.

Panwar, S. S., Rauthan, M. M. S., Barthwal, V., Mehra, N., & Semwal, A. (2024). Machine learning approaches for efficient energy utilization in cloud data centers. Procedia Computer Science, 235, 1782-1792.

Prathamesh Samal, Sagar Jha, Raman Kumar (2022). CPU Burst Time Estimation using Machine Learning. Conference: 2022 IEEE Delhi Section Conference (DELCON).

R.T. Gomez, C.M. Bermudez, Villy K.G.C. (2020). End to End Dynamic Round Robin (E-EDRR) Scheduling Algorithm Utilizing Shortest Job First Analysis. ICCMB '20, 218-221

Sudarshan M.G., Vinutha M.R., Amrutha Lakhsmi, Nischith Gowda D.Y. (2024). Predictive Model of CPU Burst Time in Computational Grids: A Comparative Study of Machine Learning Approaches. World Journal of Engineering Research and Technology.

Wang, H., Dai, YQ., Yu, J. *et al.* Predicting running time of aerodynamic jobs in HPC system by combining supervised and unsupervised learning method. *Adv. Aerodyn.* **3**, 22 (2021). https://doi.org/10.1186/s42774-021-00077-8

Wei, X. (2020). Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing. Journal of Ambient Intelligence and Humanized Computing, 1-12.

Yoga Septian, Ucuk Darusalam, Agus Iskandar (2022). Implementasi Algoritma Genetika pada Perancangan Aplikasi Penjadwalan Instalasi Antivirus Berbasis Website menggunakan Metode Waterfall. Jurnal JTIK Vol 6 No.1 Pages 125-137

Yonghao Cao, Huaqiang Yuan, Fengyao Hou, and Jianshu Hong. 2024. Regression-Classification Parallel Prediction: An Online Learning Optimization for Job Scheduling Backfill Algorithm. In *2024 8th International Conference on High Performance Compilation, Computing and Communications (HP3C 2024), June 07--09, 2024, Guangzhou, China.* ACM, New York, NY, USA 6 Pages. **https://doi.org/10.1145/3675018.3675775**