Application of LightGBM Algorithm in Production Scheduling Optimization on Non-Identical Parallel Machines

Khalid Ait Ben Hamou
Computer Sciences Engineering Laboratory, Faculty of Sciences, Cadi Ayyad University, Marrakech, Morocco
khalid.aitbenhamou@ced.uca.ma (corresponding author)

Zahi Jarir
Computer Sciences Engineering Laboratory, Faculty of Sciences, Cadi Ayyad University, Marrakech, Morocco
jarir@uca.ac.ma

Selwa Elfirdoussi
Emines - University Mohammed VI Polytechnic, Benguerir, Morocco
selwa.elfirdoussi@emines.um6p.ma

Received: 20 August 2024 | Revised: 15 September 2024 | Accepted: 19 September 2024

 DOI: https://doi.org/10.48084/etasr.8779

ABSTRACT

Production scheduling plays a decisive role in supply chain management, directly influencing the operational efficiency and competitiveness of companies. This study explores the effectiveness of the LightGBM algorithm for production scheduling on non-identical parallel machines, comparing it to algorithms such as logistic regression, KNN, decision tree, and XGBoost. LightGBM was chosen for its speed of execution and its ability to handle large amounts of data. The results show that LightGBM outperforms the other models in terms of RMSE, MAE, explained variance score, and R^2 score for regression tasks, as well as in classification accuracy for certain features. Its superiority is attributed to its ability to efficiently handle data complexity while reducing computational complexity through its leaf tree growth technique. This study highlights LightGBM's potential for improving the efficiency of supply chain management systems and the challenges associated with computational scalability for large datasets. The results suggest that LightGBM is a robust and effective solution to optimize production scheduling, paving the way for future research in this field.

Keywords-production scheduling; parallel machines; machine learning; LightGBM; optimization

I. INTRODUCTION

Production scheduling plays an important role in supply chain management and has a direct impact on operational efficiency and competitiveness. In a production environment characterized by the multiplicity of products and the diversity of non-identical parallel machines, the complexity of scheduling increases exponentially. Traditional solutions, while effective in specific contexts, often struggle to adapt to the dynamics and rapid variations of modern industrial environments. Among the various types of scheduling problems, Parallel Machine Scheduling (PMS) [1] is particularly relevant. PMS refers to the assignment of tasks to a set of parallel machines with the aim of optimizing various objectives such as total production time, costs, or resource utilization. This type of problem is commonly encountered in industrial environments where several machines of different capacities are working in parallel to process various sets of tasks. The inherent complexity of PMS calls for sophisticated approaches to find optimal or near-optimal solutions.

The main objective of this study is to explore and compare the performance of the Light Gradient Boosting Machine (LightGBM) Machine Learning (ML) algorithm with logistic regression, KNN, decision tree classification, and XGBoost. LightGBM was chosen due to its advantages in terms of execution speed and the ability to handle large amounts of data, which is crucial in production contexts with strict time requirements [2, 3]. This comparison aims to determine whether LightGBM can offer significant improvements in terms of classification accuracy and production cost prediction while optimizing the allocation of products to available machines. Testing new algorithms is essential to improve the performance of production scheduling systems. Recent advances in ML offer opportunities to develop more robust and adaptive models that are capable of better managing the variability and complexity of production data. By introducing LightGBM in this context, this study aims to provide innovative and more efficient solutions for supply chain managers, contributing to improved operational performance and reduced production costs.

Previous work on parallel machine scheduling has shown significant advances through the application of ML techniques and optimization algorithms. In [4], a hybrid learning-based meta-heuristic algorithm was developed for scheduling additive manufacturing systems with parallel SLM machines. This study used neural networks to predict processing times and optimization algorithms, such as NSGA-II and SPEA2, to assign tasks to machines. This approach demonstrated superior performance over existing methods, optimizing the coverage and distribution of solutions on the Pareto front. In [5], ML and inverse optimization were used to estimate weighting factors in multi-objective production scheduling problems. This study showed that integrating ML into optimization models better captures decision-makers' preferences and results in solutions that are more aligned with real production objectives. In [6], a deep reinforcement approach was introduced, based on Recurrent Neural Networks (RNN), to solve the scheduling problem of parallel machines with due dates and family configurations. This method modeled the problem as a Markov decision process and used Gated Recurrent Units (GRUs) to approximate the agent's policy. In [7], learning-augmented heuristics were introduced to schedule parallel machines. This approach combines supervised learning techniques with traditional heuristics to improve the accuracy of predictions and the efficiency of scheduling solutions. This combination enabled better management of the variations and uncertainties in the production data.

In [8], artificial neural networks were used to enhance multi-start grid search in serial batch scheduling problems. The ML Enhanced Grid Search (MLGS) approach predicted the best parameter configurations for the BATCS-b heuristic, significantly reducing computation times while maintaining competitive solution quality. This method optimized task assignment to machines and reduced total weighted tardiness, demonstrating the effectiveness of ML in improving traditional scheduling heuristics [8]. In [9], a model based on Deep Reinforcement Learning (DRL), called DPMS, was used to treat parallel machine scheduling problems by formulating them as a Markov Decision Process (MDP) problem. DPMS uses dispatching rules as actions and dynamically adapts them according to the environment or unexpected events, enabling efficient and adaptive rescheduling. Experimental results showed that this approach can produce promising results in dynamic environments. In [10], ML methods were used to estimate processing times in parallel machine scheduling problems. The neural network-based approach showed a significant improvement in estimation accuracy over traditional methods, leading to more efficient scheduling and reduced production times. In [11], an approach was proposed that used classification models to assign products to suitable machines and regression models to predict overall production cost. The XGBoost model stood out for its superior performance, demonstrated by reduced accuracy scores and Root Mean Squared Error (RMSE) values. In [12], a data mining method was proposed for industrial big data to solve the problem of scheduling large-scale parallel machines.

These studies show the diversity and effectiveness of ML-based approaches to parallel machine scheduling problems. However, they have revealed several important challenges. Among these, overfitting and extended training time [4] are major concerns, especially when using complex models such as deep neural networks. In addition, the complexity of determining precise weighting factors in multi-objective programming problems has been highlighted, making it difficult to obtain optimal solutions [5]. Scalability to larger problems is another notable challenge, which limits the applicability of the methods to larger real-world scenarios [8]. Furthermore, the complexity of integrating different heuristic and metaheuristic approaches into scheduling algorithms increases the implementation difficulties [7]. Finally, model accuracy under variable conditions remains an obstacle [10], as performance can fluctuate depending on variations in input data. Building on this work and attempting to address a few challenges, this study aims to explore the application of the LightGBM algorithm to optimize production scheduling, providing a detailed comparison with existing algorithms.

II. METHODOLOGY

This study uses the same dataset and preprocessing techniques as those described in [11]. This dataset contains 25710 instances of the problem of scheduling six products on three machines over a 12-period horizon, with their optimal solutions. The dataset has been normalized and standardized to ensure better performance for the ML models. This study used LightGBM to compare it with the algorithms used in [11], namely logistic regression, KNN, decision tree classification, and XGBoost. LightGBM is a boosting algorithm, based on decision trees, designed to be efficient in terms of memory and speed, and is capable of handling large amounts of data while offering high accuracy [13]. LightGBM uses a leaf tree growth technique, which reduces computational complexity and improves performance in terms of training time [14].

The metrics used to assess the performance, accuracy, and efficiency of regression models were RMSE [15], R^2 score [16], Mean Absolute Error (MAE) [15], and explained variance score. For classification, accuracy, precision, recall, and F1 score were used. [17]. These techniques and metrics were used to thoroughly evaluate and compare the performance of LightGBM against the algorithms used in [11], highlighting potential improvements in production scheduling optimization.

III. EXPERIMENTAL RESULTS

A. Regression Results

This section presents the experimental results obtained using the LightGBM algorithm and compares them with those of the algorithms used in [11], namely linear regression, KNN (k=11), decision tree, and XGBoost. Table I presents the performance of these different algorithms.

TABLE I. PERFORMANCE COMPARISON OF ML ALGORITHMS FOR PRODUCTION SCHEDULING

Metrics used	Previous algorithms [11]				Current algorithm
	KNN regression	Decision tree	Linear regression	XGBoost regression	LightGBM regression
RMSE	2900.80	1676.67	1415.74	1090.16	1074.53
MAE	2242.57	1099.49	880.53	707.51	637.57
Explained variance	0.74	0.91	0.94	0.96	0.97
R² Score	0.74	0.91	0.94	0.96	0.97

The results show that LightGBM achieved the best overall performance, with an RMSE of 1074.53, an MAE of 637.57, an explained variance score of 0.97, and an R^2 score of 0.97. These results indicate an improvement over the XGBoost model, which showed superior performance in [11], as shown in Figure 1.

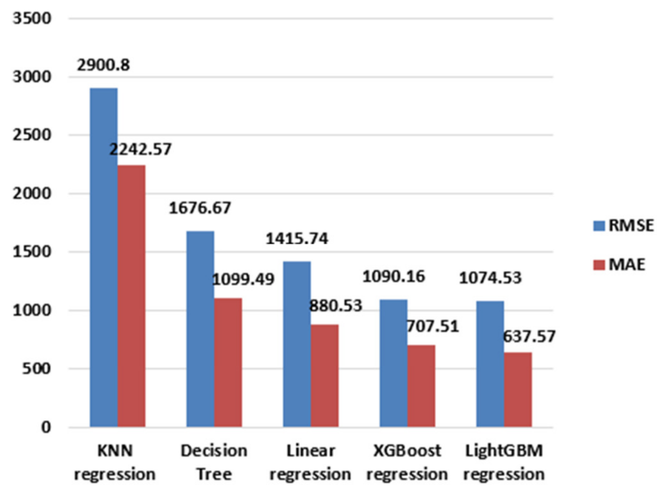

Fig. 1. Comparison of ML algorithms' performance in terms of RMSE and MAE for production scheduling.

The results show that LightGBM outperformed all other models, including XGBoost, which was the best model in [11]. This suggests that LightGBM is particularly effective for production scheduling problems on non-identical parallel machines. These experimental results confirm LightGBM's effectiveness and demonstrate its potential to improve production scheduling optimization, offering more accurate and efficient solutions for supply chain management.

B. Classification Results

This section shows the results of LightGBM compared with the algorithms in [11]. For the P1 feature, Table II shows that the XGBoost and LightGBM models performed the best in terms of accuracy, achieving a value of 0.96, closely followed by KNN and logistic regression with an accuracy of 0.92. In terms of precision, KNN slightly outperformed the other models with a score of 0.96, but had relatively low recall at 0.5, indicating difficulty in correctly identifying all positive instances. In contrast, XGBoost and LightGBM offered a better compromise with F1 scores of 0.84 and 0.85, respectively, suggesting better overall performance for this feature.

TABLE II. RESULTS FOR P1 FEATURE

Metrics used	Previous algorithms [11]				Current algorithm
	KNN	Decision tree	Logistic regression	XGBoost	LightGBM
Accuracy	0.92	0.91	0.92	0.96	0.96
Precision	0.96	0.72	0,77	0.89	0.9
Recall	0.5	0.77	0.6	0.8	0.81
F1-Score	0.48	0.74	0.63	0.84	0.85

The ROC curve in Figure 2 shows the excellent performance of the LightGBM model for the P1 feature, with an Area Under the Curve (AUC) of 0.97. This curve, close to the upper left-hand corner, indicates the high capacity of the model to correctly distinguish classes, with a high rate of true positives for a low rate of false positives.

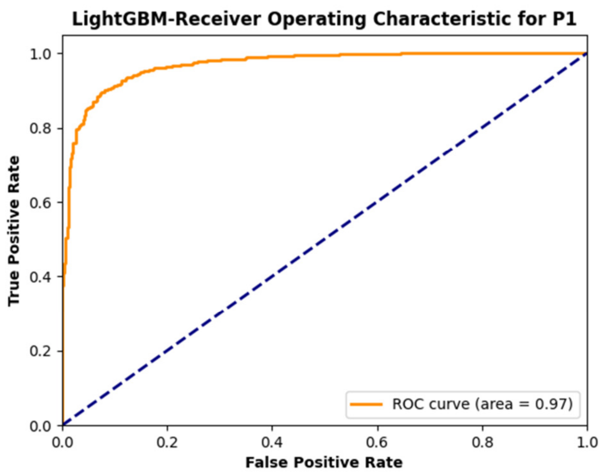

Fig. 2. Receiver Operating Characteristic (ROC) for P1.

Figure 3 depicts the calibration curve for the P1 feature, showing that the LightGBM model is well calibrated, with the curve points close to the diagonal line, indicating that the probabilities predicted by the model correspond well to the observed fractions of positives. This match between model predictions and observed results reinforces the reliability of LightGBM's predicted probabilities for the P1 feature. Figure 4 illustrates the confusion matrix for the P1 feature, showing that the LightGBM model correctly classifies the majority of samples, with 4,657 true positives and 273 true negatives. However, there are still classification errors, with 158 false positives and 54 false negatives. These results indicate a strong overall performance for the model, particularly in predicting positive cases, although there is still room for improvement in reducing false positives and false negatives. Figure 5 shows the precision-recall curve for the LightGBM model on feature P1, showing that the model maintains high precision over a wide recall range, with a slight decrease only as the recall approaches 1. This indicates that the model is effective in identifying a high proportion of true positives while minimizing false positives. The slight drop in precision at very high recall levels suggests an increase in errors, although this remains moderate. Overall, the curve demonstrates that the LightGBM model offers robust performance for the P1 feature.

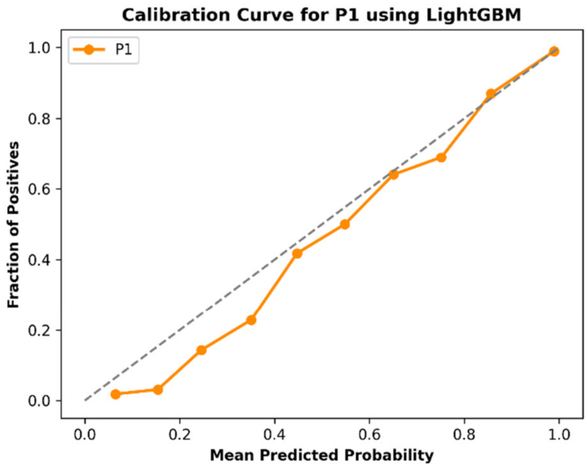

Fig. 3. Calibration curve for P1 using LightGBM.

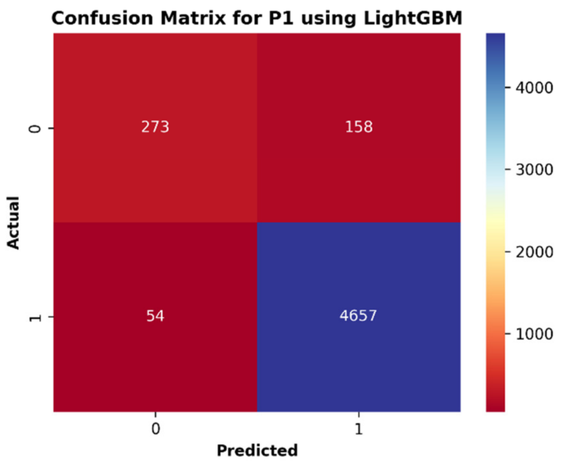

Fig. 4. Confusion Matrix for P1 using LightGBM.

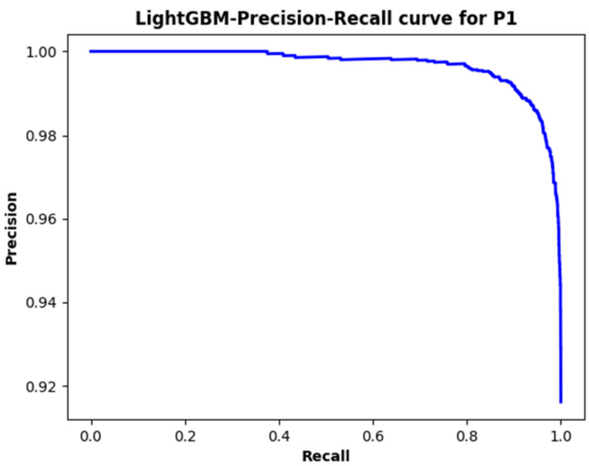

Fig. 5. Precision-Recall curve for P1 using LightGBM.

For P2, the results in Table III show that LightGBM still stands out with the best accuracy at 0.93, followed by XGBoost at 0.92. However, the accuracy of all models is significantly lower than for P1, reflecting weaker performance for this feature. The highest F1 score was achieved by LightGBM at 0.68, which, although superior to the other models, highlights the challenges faced when classifying P2, where even the best models struggle to achieve robust performance.

TABLE III. RESULTS FOR FEATURE P2

Metrics used	Previous algorithms [11]				Current algorithm
	KNN	Decision tree	Logistic regression	XGBoost	LightGBM
Accuracy	0.88	0.87	0.89	0.92	0.93
Precision	0.63	0.57	0.64	0.75	0.79
Recall	0.33	0.55	0.45	0.61	0.62
F1-Score	0.31	0.56	0.49	0.67	0.68

For feature P3, Table IV shows that all models achieved perfect performance, with accuracy, precision, recall, and F1 score of 1. This suggests that P3 is a particularly easy feature to classify, as the structured data show that P3 can only be produced by the m3 machine.

TABLE IV. RESULTS FOR FEATURE P3

Metrics used	Previous algorithms [11]				Current algorithm
	KNN	Decision Tree	Logistic Regression	XGBoost	LightGBM
Accuracy	1	1	1	1	1
Precision	1	1	1	1	1
Recall	1	1	1	1	1
F1-Score	1	1	1	1	1

For P4, Table V shows that XGBoost and LightGBM stand out with an accuracy of 0.92, slightly outperforming the other models. However, LightGBM's precision (0.79) is significantly higher than that of the others. The F1 score follows this trend, indicating that LightGBM and XGBoost are better suited to handle this feature.

TABLE V. RESULTS FOR FEATURE P4

Metrics used	Previous algorithms [11]				Current algorithm
	KNN	Decision Tree	Logistic Regression	XGBoost	LightGBM
Accuracy	0.87	0.76	0.88	0.92	0.92
Precision	0.29	0.44	0.64	0.76	0.79
Recall	0.33	0.49	0.46	0.62	0.63
F1-Score	0.31	0.46	0.49	0.67	0.69

Table VI shows the results for P5, confirming that LightGBM again excels with an accuracy of 0.99, closely followed by XGBoost at 0.98. These models also show high precisions, 0.98 for LightGBM and 0.94 for XGBoost, suggesting high reliability in predicting positive classes. LightGBM's F1 score is the highest at 0.89, indicating an overall superior performance, particularly in terms of the trade-off between precision and recall. Finally, for P6, Table VII indicates that LightGBM and XGBoost continue to show their superiority with an accuracy of 0.99. LightGBM stands out with a precision of 0.99 and a recall of 0.8, resulting in an F1 score of 0.87. These results confirm the trend observed on the other features, where LightGBM and XGBoost are consistently the best-performing models, offering an optimal balance between the various performance metrics, while the other models show more varying results.

TABLE VI. RESULTS FOR FEATURE P5

Metrics used	**Previous algorithms [11]**				**Current algorithm**
	KNN	**Decision tree**	**Logistic regression**	**XGBoost**	**LightGBM**
Accuracy	0.96	0.97	0.96	0.98	0.99
Precision	0.48	0.8	0.75	0.94	0.98
Recall	0.5	0.79	0.52	0.81	0.82
F1-Score	0.49	0.79	0.53	0.86	0.89

TABLE VII. RESULTS FOR FEATURE P6

Metrics used	**Previous algorithms [11]**				**Current algorithm**
	KNN	**Decision Tree**	**Logistic Regression**	**XGBoost**	**LightGBM**
Accuracy	0.97	0.98	0.97	0.99	0.99
Precision	0.48	0.8	0.68	0.98	0.99
Recall	0.5	0.85	0.52	0.79	0.8
F1-Score	0.49	0.83	0.53	0.86	0.87

Comparing LightGBM and XGBoost on all six features, it is clear that the two models perform very similarly, with the former being superior in most cases. For features P1, P2, P4, P5, and P6, LightGBM outperforms XGBoost in terms of precision and F1 score, suggesting a better ability to minimize false positives while maintaining good recall. For example, for P5, LightGBM displays a precision of 0.98 versus 0.94 for XGBoost and an F1 score of 0.89 versus 0.86, indicating better overall performance. Similarly, for the P2 feature, although the gap is smaller, LightGBM manages to achieve a slightly higher F1 score of 0.68 versus 0.67 for XGBoost. Overall, although the differences are sometimes small, LightGBM seems to offer a marginal advantage over XGBoost in terms of precision and the balance between precision and recall, particularly for more complex features such as P1 and P5.

IV. DISCUSSION

The regression results show that the integration of the LightGBM algorithm brought significant improvements over the algorithms used in [11], as it outperformed them in terms of RMSE, MAE, explained variance score, and R^2, confirming its robustness and effectiveness for production scheduling on non-identical parallel machines. LightGBM's superiority can be attributed to its ability to efficiently handle large amounts of data while optimizing prediction accuracy, thanks to its leaf tree growth technique that reduces computational complexity.

The classification results also show some interesting trends. For the P1 feature, LightGBM performed slightly better than XGBoost, with slightly better accuracy and F1 score. This suggests that LightGBM can be slightly more effective in contexts where the performance of boosting models is already high. On the other hand, for the P2 feature, LightGBM also outperformed XGBoost, although the difference was less, which could indicate that LightGBM performs better for more complex or less well-separated features. For feature P3, all models achieved perfect performance, indicating easy class separation for this feature. For features P4, P5, and P6, LightGBM maintained slightly better performance than XGBoost, confirming its tendency to outperform other models in complex classification situations. This ability to deliver high performance, even with diverse datasets, reinforces the idea that LightGBM is a robust choice for classification applications in addition to regression tasks.

Although LightGBM and XGBoost show remarkable performance, computational scalability remains a challenge for very large datasets. Adopting approximation strategies or heuristics may be necessary to maintain optimal performance while reducing computational costs. Future studies could also explore the integration of real-time feedback and dynamic adjustments to make production systems even more responsive and economically efficient.

Several interesting perspectives can be envisaged to extend this work. It would be relevant to explore the fusion of different ML strategies [18] and other ML algorithms or ensemble techniques to compare and potentially improve current performance. Applying this approach to other industrial fields would allow us to test its universality. A study of the costs and benefits of implementing this model in real production would be crucial to assess its economic impact. Finally, analysis of the model's robustness in the face of imperfect data, as well as ethical and sustainability considerations, deserves particular attention to ensure responsible and sustainable adoption of these technologies.

V. CONCLUSION

This study demonstrated the effectiveness of the LightGBM algorithm for optimizing production scheduling on non-identical parallel machines. Experimental results showed that LightGBM outperformed other models in terms of RMSE, MAE, explained variance score, and R^2 score, highlighting its ability to handle large amounts of data and deliver accurate predictions. The potential impact of using LightGBM to improve the efficiency of supply chain management systems is significant, offering more accurate predictions and optimized task assignments. The limitations observed, particularly regarding computational scalability for large datasets, suggest the need for future research. Such research could focus on adopting approximation strategies, optimizing hyperparameters, and exploring deep learning techniques. Integrating real-time feedback and dynamic adjustments into models could also make production systems more responsive and economically efficient.

In short, this study confirms the robustness and effectiveness of the LightGBM algorithm for production scheduling and paves the way for further research into the optimization of supply chain management systems using advanced ML techniques.

REFERENCES

[1] K. C. Ying, P. Pourhejazy, and X. Y. Huang, "Revisiting the development trajectory of parallel machine scheduling," *Computers & Operations Research*, vol. 168, Aug. 2024, Art. no. 106709, https://doi.org/10.1016/j.cor.2024.106709.

[2] M. Alanazi, R. S. Aldahr, and M. Ilyas, "Human Activity Recognition through Smartphone Inertial Sensors with ML Approach," *Engineering,*

Technology & Applied Science Research, vol. 14, no. 1, pp. 12780–12787, Feb. 2024, https://doi.org/10.48084/etasr.6586.

[3] L. Li, Z. Liu, J. Shen, F. Wang, W. Qi, and S. Jeon, "A LightGBM-based strategy to predict tunnel rockmass class from TBM construction data for building control," *Advanced Engineering Informatics*, vol. 58, Oct. 2023, Art. no. 102130, https://doi.org/10.1016/j.aei.2023.102130.

[4] M. Rohaninejad, R. Tavakkoli-Moghaddam, B. Vahedi-Nouri, Z. Hanzálek, and S. Shirazian, "A hybrid learning-based meta-heuristic algorithm for scheduling of an additive manufacturing system consisting of parallel SLM machines," *International Journal of Production Research*, vol. 60, no. 20, pp. 6205–6225, Oct. 2022, https://doi.org/10.1080/00207543.2021.1987550.

[5] H. Togo, K. Asanuma, T. Nishi, and Z. Liu, "Machine Learning and Inverse Optimization for Estimation of Weighting Factors in Multi-Objective Production Scheduling Problems," *Applied Sciences*, vol. 12, no. 19, Jan. 2022, Art. no. 9472, https://doi.org/10.3390/app12199472.

[6] F. Li, S. Lang, B. Hong, and T. Reggelin, "A two-stage RNN-based deep reinforcement learning approach for solving the parallel machine scheduling problem with due dates and family setups," *Journal of Intelligent Manufacturing*, vol. 35, no. 3, pp. 1107–1140, Mar. 2024, https://doi.org/10.1007/s10845-023-02094-4.

[7] A. Uzunoglu, C. Gahm, S. Wahl, and A. Tuma, "Learning-augmented heuristics for scheduling parallel serial-batch processing machines," *Computers & Operations Research*, vol. 151, Mar. 2023, Art. no. 106122, https://doi.org/10.1016/j.cor.2022.106122.

[8] A. Uzunoglu, C. Gahm, and A. Tuma, "A machine learning enhanced multi-start heuristic to efficiently solve a serial-batch scheduling problem," *Annals of Operations Research*, vol. 338, no. 1, pp. 407–428, Jul. 2024, https://doi.org/10.1007/s10479-023-05541-w.

[9] C. L. Liu, C. J. Tseng, T. H. Huang, and J. W. Wang, "Dynamic Parallel Machine Scheduling With Deep Q-Network," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 11, pp. 6792–6804, Aug. 2023, https://doi.org/10.1109/TSMC.2023.3289322.

[10] H. Yamashiro and H. Nonaka, "Estimation of processing time using machine learning and real factory data for optimization of parallel machine scheduling problem," *Operations Research Perspectives*, vol. 8, Jan. 2021, Art. no. 100196, https://doi.org/10.1016/j.orp.2021.100196.

[11] K. A. B. Hamou, Z. Jarir, and S. Elfirdoussi, "Design of a Machine Learning-based Decision Support System for Product Scheduling on Non Identical Parallel Machines," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 16317–16325, Oct. 2024, https://doi.org/10.48084/etasr.7934.

[12] Y. Li *et al.*, "A K-means-Teaching Learning based optimization algorithm for parallel machine scheduling problem," *Applied Soft Computing*, vol. 161, Aug. 2024, Art. no. 111746, https://doi.org/10.1016/j.asoc.2024.111746.

[13] G. Ke *et al.*, "LightGBM: a highly efficient gradient boosting decision tree," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, CA, USA, Sep. 2017, pp. 3149–3157.

[14] D. Wang, L. Li, and D. Zhao, "Corporate finance risk prediction based on LightGBM," *Information Sciences*, vol. 602, pp. 259–268, Jul. 2022, https://doi.org/10.1016/j.ins.2022.04.058.

[15] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geoscientific Model Development*, vol. 15, no. 14, pp. 5481–5487, Jul. 2022, https://doi.org/10.5194/gmd-15-5481-2022.

[16] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Computer Science*, vol. 7, Jul. 2021, Art. no. e623, https://doi.org/10.7717/peerj-cs.623.

[17] Ž. Đ. Vujovic, "Classification Model Evaluation Metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021, https://doi.org/10.14569/IJACSA.2021.0120670.

[18] S. Nuanmeesri, "A Hybrid Deep Learning and Optimized Machine Learning Approach for Rose Leaf Disease Classification," *Engineering, Technology & Applied Science Research*, vol. 11, no. 5, pp. 7678–7683, Oct. 2021, https://doi.org/10.48084/etasr.4455.