Applied Artificial Intelligence and Computing Systems
https://www.scigate.org
Science Gate Academic
Vol. 1, No. 1, 2025

Article

Intelligent Scheduling Optimization for Flexible Manufacturing Systems

Ansel Kittredge
Middle Tennessee State University, Murfreesboro, USA
akittredge@mtsu.edu

Abstract: *This study addresses the challenge of dynamic processing time prediction in intelligent manufacturing systems by proposing a Transformer-based time series prediction model enhanced with prior knowledge weighting. To validate the approach, an intelligent processing unit experimental platform was designed, integrating processing, logistics, and perception modules to simulate flexible manufacturing scenarios. The proposed model employs multi-head attention mechanisms and weighted feature optimization to capture complex temporal dependencies within processing sequences, emphasizing the influence of key materials in each scheduling cycle. Experimental results demonstrate that the weighted Transformer model achieves superior performance compared to conventional neural network architectures, with a coefficient of determination (R^2) of 0.989 and significantly reduced mean absolute and root mean squared errors. The results confirm the model's ability to adapt to time-varying processing conditions, improving scheduling accuracy and overall production efficiency in intelligent manufacturing environments.*

Keywords: *Intelligent manufacturing; dynamic scheduling; Transformer model*

1. Introduction

With the rapid development of intelligent manufacturing technologies, intelligent processing units [1], as core enablers of Industry 4.0 [2], are becoming pivotal in achieving flexible production and dynamic scheduling. In complex and volatile manufacturing scenarios, how to optimize production efficiency and reduce resource waste through precise scheduling has emerged as a focal point of common concern in both academia and industry [3]. Traditional scheduling systems predominantly rely on fixed processing time assumptions, essentially employing static rules (such as shortest processing time first) for linear programming of production tasks [4]. However, such methods struggle to adapt to dynamically changing processing requirements in flexible production environments [5]: On one hand, material processing times exhibit significant time-varying characteristics due to multiple influencing factors including equipment status, human operations, and environmental disturbances; On the other hand, in small-batch customized production scenarios featuring diverse material types and complex processing sequences [6], traditional models fail to effectively capture temporal dependencies and dynamic coupling features [7]. Scheduling systems based on fixed processing time assumptions demonstrate substantial average errors in flexible production lines, leading to increased equipment idle rates and order delivery delays.

This study focuses on addressing dynamic processing time prediction in intelligent processing unit scheduling systems. To overcome the limitations of fixed-time assumptions in conventional approaches, we propose a Transformer-based time series prediction model enhanced with prior knowledge weighting. The effectiveness of the proposed model is rigorously trained and validated through a self-developed intelligent processing unit experimental platform.

2. Intelligent Processing Units

2.1 .Deasign of the Intelligent Processing Units

In intelligent scheduling research, the intelligent processing unit serves as a critical experimental platform, necessitating systematic design of its physical implementation [8]. As illustrated in Figure 1, the experimental platform comprises three functionally integrated modules: processing module, logistics module, and perception module, corresponding to processing, logistics, and sensing operations respectively.

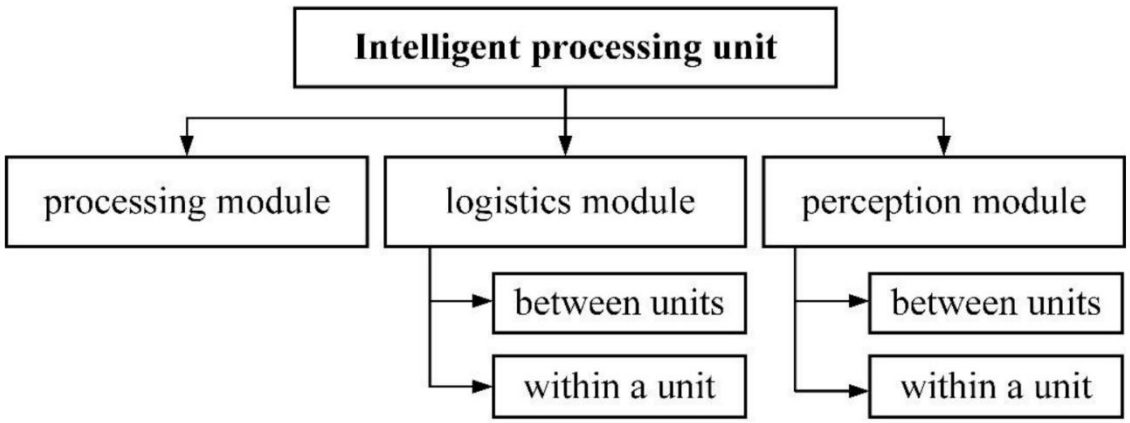

Figure 1. Schematic diagram of the intelligent processing unit experimental platform

The processing module is the main body of the intelligent processing unit that processes materials and increases their added value. The design of this part meets the requirements for production equipment in the scheduling system. The logistics module primarily enables the circulation of processed materials within the scheduling system. It not only includes material distribution within the intelligent processing unit but also facilitates material exchange with upstream and downstream intelligent processing units. This module mainly consists of auxiliary equipment in the scheduling system. The perception module is designed to enable state awareness of equipment and environments for the scheduling system. The implementation of this module involves not only detection devices in the scheduling system but also various sensors integrated into production equipment and auxiliary equipment. Sensors embedded in production/auxiliary equipment, combined with additional sensors, form the hardware foundation of the perception module.

2.2 Establishment of the Intelligent Processing Units

To realize the intelligent processing unit with the aforementioned functionalities, this section focuses on the equipment for processing and material handling functions. The devices of the perception module are integrated into these two categories of equipment and require no separate discussion. For scheduling system research based on the intelligent processing unit, collaborative robots are selected as the primary processing equipment. A buffer zone is established, with two conveyors serving as main devices to implement dual buffering for input and output material flow in the intelligent processing unit.

To achieve the required processing capabilities, collaborative robots equipped with end- effectors are adopted as production equipment. With rapid advancements in AI technologies, breakthroughs in large-scale models and embodied intelligence continuously enhance collaborative robots' autonomous learning and scene generalization capabilities. As increasing technological applications converge, collaborative robots are positioned as critical productivity tools in intelligent manufacturing visions, hence their selection as the production equipment for the experimental platform.

To enable internal scheduling decisions within the intelligent processing unit, conveyors integrated with multi-modal sensing devices are chosen as loading/unloading interfaces for the logistics system, collectively forming the buffer equipment. The distinction between loading/unloading interfaces depends on the conveyor's functional role in the unit. For studying scheduling decision methods, the buffer capacity is designed to hold 10 workpieces, with 5 allocated to each interface. Within this capacity, the scheduling system may select any workpiece from the loading interface. However, the replenishment position for the loading interface is fixed, meaning newly arrived materials always occupy the outermost position of the unit. To address this constraint, a material relocation strategy is developed.

The design assigns Position 1 to the workpiece closest to the processing location, with numbering incrementing to Position 5 as distance increases. As shown in Figure 2, materials are transported from Position 5 at the feeding port via the conveyor and finally placed at Position 1. The critical principle of the adjustment strategy is to minimize conveyor motion complexity: ensuring Position 1 remains material-free during conveyor movement. If the scheduling result selects material at Position 1 for processing, no material relocation is required; otherwise, relocation is mandatory. When the conveyor is not fully loaded, a position transfer strategy is triggered: materials at Position 1 are moved to the smallest-numbered vacant position to maintain Position 1 empty, facilitating conveyor advancement. After conveyor advancement, Position 5 is inevitably emptied, allowing new materials to be accepted.

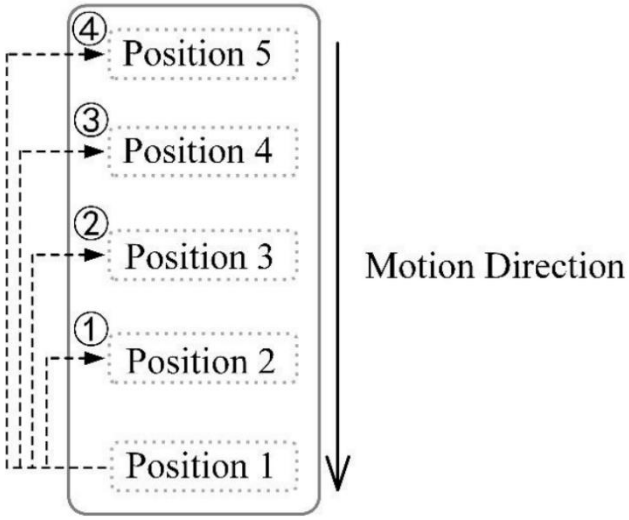

Figure 2. Relocation strategy

In the left part of Figure 2, dashed line ① corresponds to the position transfer strategy when the selected workpiece for processing is at Position 2. Specifically, after the material at Position 2 is removed, the material at Position 1 is transferred to Position 2. Dashed lines ②, ③, and ④ follow the same logic. After completing the relocation strategy, the conveyor can be controlled to advance. Infrared sensors can only detect the presence/absence of materials at defined positions but cannot identify material types. Therefore, a depth camera is installed directly above Position 5 to recognize material categories.

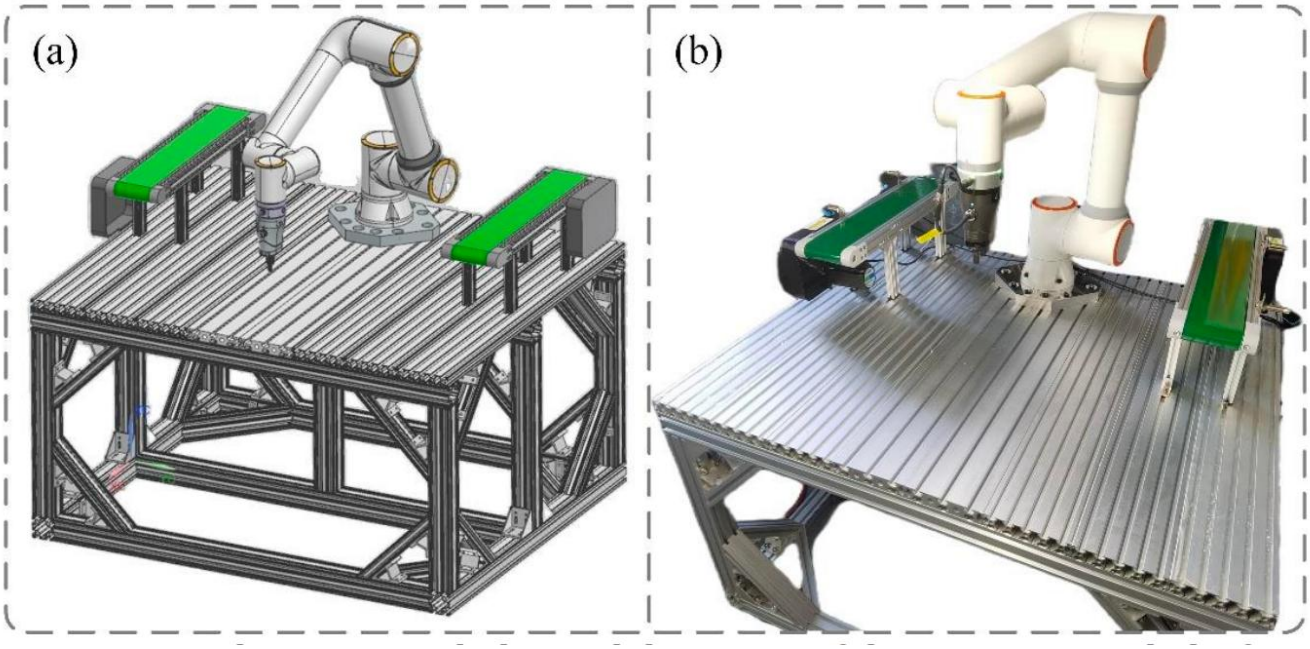

Figure 3. Design drawings and physical diagrams of the experimental platform

Based on the architectural design of the intelligent processing unit scheduling system, the experimental platform requires integration of primary production equipment including collaborative robots and conveyors. Considering the need for physical support during equipment installation and simulated production phases, a physical platform must be designed as the base structure of the experimental platform. Device integration and experimental design/planning are implemented on this base. The design and assembly result of the complete platform are shown in Figure 3, where two diffuse photoelectric sensors are equipped on one side of each conveyor in the physical diagram.

After completing equipment selection for the experimental platform guided by functional requirements, data communication design must be implemented. The design focuses on collaborative robots, conveyors, sensors, and depth cameras. These device categories cover all types in the aforementioned equipment conFigureuration, with multi-source heterogeneous data characteristics requiring distinct communication protocols. Specific implementation methods are detailed in Figure 4.

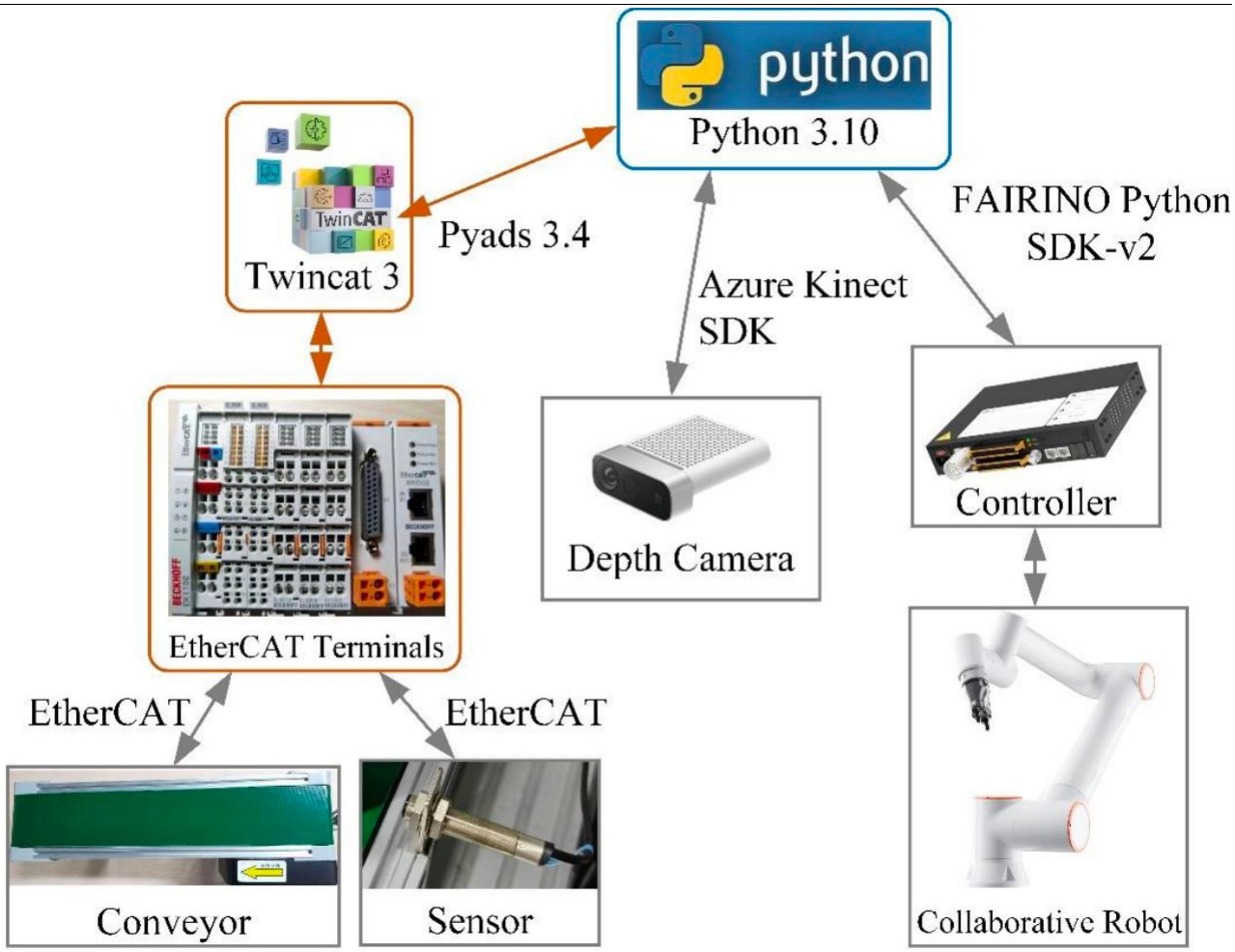

Figure 4. Communication diagram of the experimental platform

3. Processing Time Prediction Model

3.1 Processing Time Prediction Requirements

The scheduling system of the intelligent processing unit requires not only material identification but also associated processing information, such as processing time. In traditional scheduling methods, to simplify computational processes, the processing time of specific materials at designated workstations is assigned fixed values. However, this assumption only holds for rigid production lines. When flexible production modes are implemented or human operator influences are considered, processing time inevitably becomes variable.

Current statistical approaches for processing time exhibit limitations, primarily relying on aggregating all historical machining durations for statistical processing. However, production processes constitute continuous actions where operations at any given time node inherently influence preceding and subsequent events. This interdependence is particularly critical in highly integrated intelligent processing units, where equipment interactions induce time-series variations that must be accounted for. Therefore, a complete processing sequence is proposed as Figure 5, with workpieces selected in each scheduling decision cycle marked by blue ellipses, accompanied by their actual machining times.

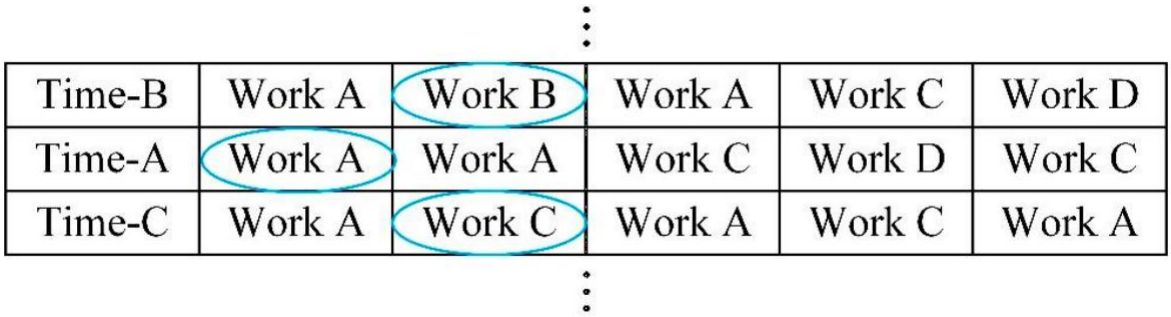

Figure 5. Complete processing sequence list

Scheduling cycles are sequentially connected end-to-end within a complete machining process, meaning the total process duration equals the sum of all individual cycles. Even if only one material is selected and processed per cycle, the cycle duration cannot be directly adopted as the material's processing time for precise characterization. Considering the intelligent processing unit's workflow, processing status is triggered when collaborative robots retrieve materials from the buffer zone. This timestamp is determined by monitoring robot status. Processing completion is registered when robots initiate transfer of processed materials back to the buffer. System timestamps captured at these two action initiations are differenced to calculate actual processing time.

For new decision cycles, the task update module acquires the latest single-cycle processing sequence. All materials in this sequence require processing time predictions, which are combined with the sequence data to serve as inputs for subsequent intelligent scheduling algorithms.

3.2 Transformer Models

From the aforementioned analysis and discussion, it is concluded that processing time prediction primarily relies on processing sequences annotated with historical time data. Observation of historical data tables reveals that processing time prediction is fundamentally based on processing sequences, where the sequence serves as the method's input and the output corresponds to predicted processing times for individual materials. Neural networks, computational models inspired by biological neural perception, initially featured simple architectures with only input, hidden, and output layers, capable of basic classification tasks. With progressive technological advancements, the Transformer model was proposed to better capture long-range dependencies in sequential data [9]. Currently widely applied in natural language processing and speech recognition domains, this architecture is considered suitable for processing time prediction given that processing sequences inherently record consecutive production processes.

The architecture of the Transformer model is shown in Figure 6, which primarily consists of several key components including positional encoding, encoder-decoder architecture, and multi-head attention mechanism.

The terms "Inputs" and "Outputs" at the bottom of the Figure. 6 refer to the training data provided to the model and the expected target output, respectively. The "Output Probabilites" at the top denotes the probabilistic distribution generated by the trained model, ultimately manifested as predictive outcomes. Taking a translation task as an example, the "Inputs" at the bottom would be English text, the "Outputs" would be the corresponding Chinese translation, and the " Output Probabilites" at the top represents the model's predictions after training. Both the English input and Chinese output sequences entering the Transformer model first undergo input and output embedding processes, converting character sequences into vector matrices.

Following this conversion, positional encoding is applied, typically implemented through calculations involving sine and cosine functions. This step integrates positional information of elements within the sequence into the original representations. The resulting augmented vectors are then processed through the encoder in the left gray box and decoder in the right gray box modules. The notation "N × " adjacent to both gray boxes indicates the potential stacking of multiple encoder and decoder layers through sequential concatenation.

Within the encoder, multi-head attention mechanisms and feed forward neural networks operate sequentially to encode English character representations. The decoder architecture bifurcates based on encoder data injection points: the first segment processes masked multi- head attention to prevent premature exposure to subsequent positional information, while the second segment operations by jointly decoding English-Chinese representations to produce final outputs. The multi-head attention mechanism at the beginning of the decoding process is masked to prevent the model from observing future information while generating the output sequence.

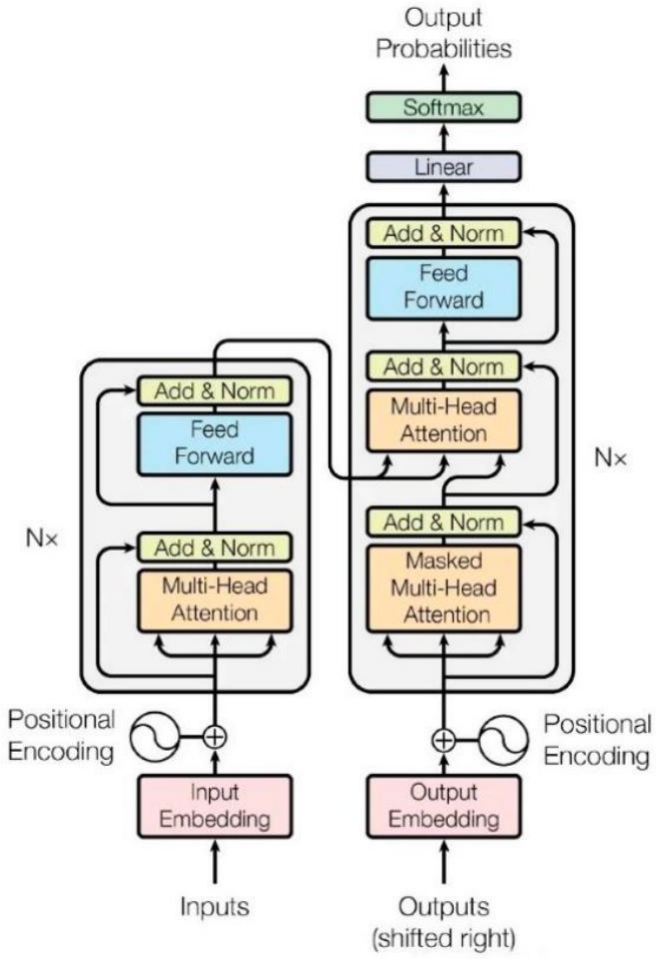

Figure 6. Transformer model [9]

The multi-head attention mechanism constitutes the most critical component within the model [10]. Its implementation involves applying three distinct linear layers: Query (Q), Key (K), and Value (V), to perform feature transformation on input data. These transformed Q, K, and V matrices are subsequently combined through scaled dot product attention computation. The mechanism achieves parallelism by

simultaneously executing multiple such combination processes, with the quantity of parallel processes designated as the number of heads (h). The outputs from all attention heads are concatenated and ultimately processed through a linear layer to produce the final output.

In summary, the Transformer model demonstrates distinct advantages in addressing temporal sequence problems, rendering it particularly suitable for processing time prediction within intelligent processing unit scheduling problems. To further optimize its application in scheduling methodologies, weighted feature optimizations can be implemented across different input characteristics, coupled with strategic incorporation of prior knowledge to enhance prediction accuracy.

3.3 Weighted Optimization Mechanism

In the comprehensive processing sequence list, processing times exhibit one-to-one correspondence with their respective processing rounds, where the Transformer model receives processing sequences as "Inputs" and generates processing time predictions as "Outputs." The model ultimately predicts material-specific processing times under varying sequence conFigureurations. As outlined in the preceding section, encoded representations of processing sequences and their corresponding times are processed through the multi-head attention mechanism. Within this framework, all materials in the processing sequence receive equal weighting. Under ideal conditions, the model could theoretically discern the relative significance of different materials and their positional relationships in influencing processing durations via the multi-head attention mechanism. However, relying on autonomous discovery of these critical relationships imposes substantial computational demands and risks failure to identify the most influential material parameters limitations that hinder practical implementation in intelligent processing unit. To address it, weighted feature optimization can be systematically implemented to amplify the contribution of dominant features while attenuating secondary ones, thereby enhancing the model's predictive accuracy.

In the scheduling system of the intelligent processing unit, the prior knowledge is mainly determined by the material processed for the corresponding processing time of a single processing sequence, that is, the prior knowledge can be interpreted [11]. In the complete processing sequence table given for training, the processing materials of every round are not marked, that is, all materials in each processing sequence are given the same weight when processed by the multi-head attention mechanism. The model will spend a considerable amount of time to find out what the most influential feature is. In order to avoid this problem, the materials selected for each scheduling cycle in the processing sequence are marked, and the model is informed in advance to transfer more attention weights to this feature.

The specific details of the feature weight are to set a weight vector ω = [1, 1, 1...], and the number of elements in the vector is consistent with the number of features. When the elements in the weight vector are equal to 1, it will not affect the corresponding feature; if it is less than 1, it will reduce the influence of the corresponding feature in the attention mechanism; if it is greater than 1, it will increase the influence of the corresponding feature. Setting all the elements of the weight vector to 1, that is, without any processing, is the standard form of the Transformer model. In addition, by setting the weight vector to nn.Parameter, the model can dynamically adjust this during training.

4. Experiments and Results

Considering the amount of data and multiple random tests, the key parameters of the model are tentatively set to be 64 for the batch size and 8 for the feature embedding dimensions, which will be adjusted appropriately according to the training results of the model. The equipment used for training is CPU: Intel i5-13400f, GPU: NVIDIA GeForce RTX 2070 SUPER, in addition, the maximum training period is set to be 300, but an additional mechanism is designed for the early stopping method, the triggering condition is that the loss of the 15 consecutive training periods is not optimized, and the training process will be terminated prematurely after the early stopping method is triggered. The early

stopping method will end the training process of the model early after triggering it. The number of hidden neurons is set to 64, the number of heads in the multi-head attention mechanism is 8, the dropout rate is 0.001, the number of encoders and decoders is 3, and the learning rate is initially set to 0.001.

Based on the weighted optimization of prior knowledge, the features are divided into primary and secondary features, where the primary feature is the material selected for processing in the corresponding scheduling cycle, and the rest of the materials are secondary features. In order to strengthen the influence of the primary feature, its weighted value is set to 10, meaning the influence of this feature will be magnified by 10 times when it is perceived in the multi-attention mechanism module. The weight of the secondary feature can also be set to 0.1 to achieve weakening the influence of the secondary feature. In the training process, a total of three weighting methods with initial ratios of 10:0.1, 10:1 and 1:1 for the primary and secondary features are considered, corresponding to the three cases of strengthening the primary feature while weakening the secondary feature, strengthening only the primary feature and no weighting, respectively.

After training, the model was evaluated using the coefficient of determination (R2) , mean absolute error regression loss (MAE), root mean squared error (RMSE) and mean absolute percentage error (MAPE) to evaluate the model [12].

Table 1 shows the effect of different batch sizes on model training, and it can be found that increasing the batch size to 128 under this dataset does not necessarily improve the performance of the model. Under the three weighting methods, there is and only the weighting ratio of 1:1 obtains the improvement. However, in the comparison of different weighting ratios, the model with 1:1 has the worst performance, and even if the performance of the model under this weighting ratio is improved by increasing the batch size, the performance is worse compared to the performance of the model with the other two weighting ratios. In this comparison, the tentative batch size of 64 can be continued for the next optimization, and it is also determined that either the weight ratio of 10:0.1 or 10:1 has an optimizing effect on the model, which verifies the reasonableness of the weighted optimization based on the a priori knowledge to a certain extent.

Table 1. Effect of the batch size

Batch size	Weight	R2	MAE	RMSE	MAPE
64	10:00.1	0.98387	1.33332	1.73059	0.0326
64	10:01	0.91914	3.1322	3.87504	0.08535
64	1:01	0.81077	4.3807	5.92795	0.088
128	10:00.1	0.91725	3.19767	3.9201	0.08416
128	10:01	0.91511	3.1282	3.97048	0.08458
128	1:01	0.91308	3.22651	4.01749	0.08743

Discussing the effects of different embedding dimensions on the model on the basis of the above experiments, it is found that increasing the embedding dimensions can comprehensively improve the performance of the model, strengthen the learning ability of the model, and be more capable of capturing the differences and changes in the processing sequence. After increasing the embedding dimension, the performance of the model with all three weight ratios is improved to some extent, and the comparison relationship between different weight ratios remains consistent, still the weight ratio of 10:0.1 is the most effective, the weight ratio of 10:1 is the second most effective, and the weight ratio of 1:1 is the most

ineffective. That is, the weighted optimization based on prior knowledge has a positive effect on the training of the model and meets the expectation.

Table 2. Effect of the embedding dimension

Batch size	Weight	R2	MAE	RMSE	MAPE
64	10:00.1	0.98387	1.33332	1.73059	0.0326
64	10:01	0.91914	3.1322	3.87504	0.08535
64	1:01	0.81077	4.3807	5.92795	0.088
128	10:00.1	0.98936	1.04327	1.4051	0.02799
128	10:01	0.9198	3.11935	3.85915	0.0843
128	1:01	0.91675	3.22449	3.93178	0.08628

Further discussing the different parameter variations, where the R2 enhancement is smaller, this is because the model's metrics have reached 0.98, which is very close to the ideal value of 1 for this parameter and is close to the conclusions drawn by other scholars using this criterion [13]. Considering the training data and the uncertainty after applying to the intelligent processing unit, based on this parameter, it can be considered that the training results of the model can already meet the needs of the intelligent processing unit scheduling system.MAE and RMSE react to the model's error and error amplitude, respectively, and have obtained a better representation of the parameters when the weight ratio is 10:0.1, which is more similar to the optimization amplitude of other scholars [14]. MAPE mainly embodies the degree of adaptation of the model for different data, the optimization results of this parameter show that the trained model can have better performance under different processing sequences. Taken together the model works best when the batch size is 64, the embedding dimension is 16, and the weight ratio is 10:0.1. The loss variation of the training process under this condition is shown in Figure 7.

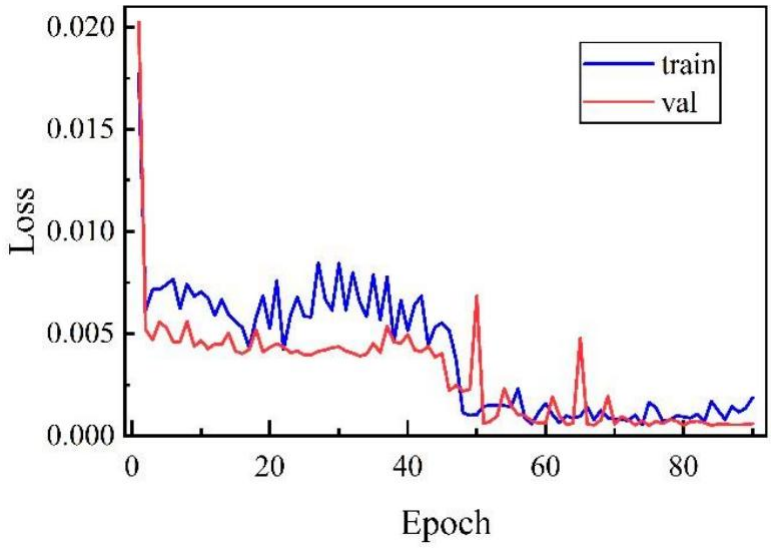

Figure 7. Curves of loss

Table 3 shows the results of the weighted optimized Transformer model compared with two other common neural network models. the CNN and the MLP, whose network performance is similar to that of the unweighted optimized Transformer model, and whose performance differs significantly from that of

the optimized Transformer model, which, to a certain extent, proves the optimization algorithms used to be effectiveness of the optimization algorithms used.

Table 3. Comparison of different models

Model	R2	MAE	RMSE	MAPE
Improved Transformer	0.98936	1.04327	1.4051	0.02799
CNN	0.92635	2.97218	3.69808	0.08051
MLP	0.90285	3.02995	4.2475	0.08887

Randomly intercept 100 rows in the test set, that is, 100 rounds of processing sequence, and pass it as input to the above mentioned optimal model. The predicted value is shown in Figure 8. It can be seen from the Figureure that the predicted processing time is not much different from the ture processing time given, which is basically in line with the processing time designed in the previous text.

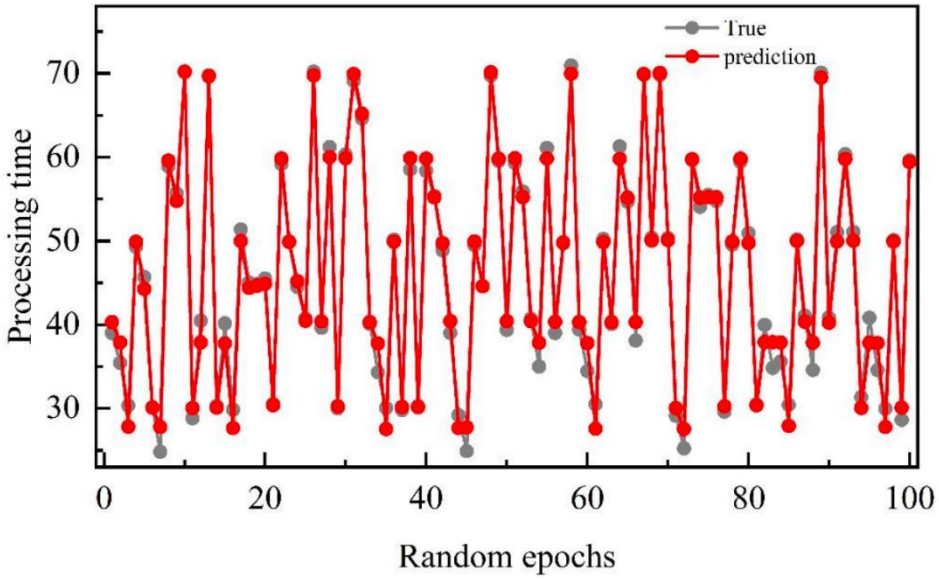

Figure 8. Comparison of processing time prediction results with ture values

5. Conclusion

Aiming at the problem of dynamic processing time prediction in the scheduling process of intelligent processing units in intelligent manufacturing systems, this paper first designs and constructs the experimental platform and processing sequence list of intelligent processing units, then optimizes the Transformer model based on prior knowledge weighting, and finally uses the processing sequence list to train the optimized model. The results show that the weighted improved Transformer model can predict the processing time with high performance, and also shows certain advantages compared with the two commonly used neural networks.

References

[1] J. Ma, Y.C. Wang, M. Zhao, et al. Visualization management and control methods of production cell based on digital twin. Computer Integrated Manufacturing Systems, 2021, 27, p.1256 – 1268. (In Chinese)

[2] Pivoto, D. G. S., de Almeida, L. F. F., Righi, R. da R., Rodrigues, J. J. P. C., Lugli, A. B., & Alberti, A. M. Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review. Journal of Manufacturing Systems, 2021, 58, p.176 – 192.

[3] M. Benzaouia, B. Hajji, A. Mellit, et al. Fuzzy-IoT smart irrigation system for precision scheduling and monitoring. Computers and Electronics in Agriculture, 2023, 215, p.108407.

[4] Mateja Đumić, Domagoj Jakobović. Using priority rules for resource-constrained project scheduling problem in static environment. Computers & Industrial Engineering, 2022, 169, p.108239.

[5] Y. Wan, T.Y. Zuo, L. Chen, et al. Efficiency-Oriented Production Scheduling Scheme: An Ant Colony System Method. IEEE Access, 2020, 8, p.19286 – 19296.

[6] X. Shi, J. Lin, M. Zheng, et al. Production Decisions for Standard and Customized Products with Postponement. IEEE International Conference on Industrial Engineering and Engineering Management 2020, Singapore, p.741 – 745.

[7] H. Sun. Manufacturing scheduling with genetic algorithm and LSTM neural networks. International Journal of Simulation Modelling, 2023, 22, p.508 – 519.

[8] S.W. Zhang, Z. Wang, D.J. Cheng, et al. An intelligent decision-making system for assembly process planning based on machine learning considering the variety of assembly unit and assembly process. The International Journal of Advanced Manufacturing Technology, 2022, 121, p.805 – 825.

[9] A. Vaswani, N. Shazeer, N. Parmar, et al. Attention Is All You Need. arXiv, 2017, arXiv:1706.03762.

[10] S.Y. Cheng, Y.G. Liu. Research on Transportation Mode Recognition Based on Multi-Head Attention Temporal Convolutional Network. Sensors, 2023, 23, p.3585.

[11] J.J. Gu, Q.K. Zhao, B.L. Yin. Lowering the sound transmission loss of impedance-matching structures: Data-driven optimization assisted with a priori knowledge. Materials & Design, 2023, 232, p.1120911.

[12] C. Fu, K.J. Wang. LASSO Regression Assisted by Coefficient of Determination and Correlation Coefficient. Journal of Fujian Normal University (Natural Science Edition), 2024, 40, p.57 – 63+89. (In Chinese)

[13] Z.C. Sun, F.Y. Zhou, T.L. Dai, et al. Evaluation Method for Estimation Accuracy of Battery Electric Vehicle Cruising Range. Science Technology and Engineering, 2019, 19, p.344 – 350. (In Chinese)

[14] S.Y. Cai, W.T. Kuang. Aeroengine baseline prediction model based on improved ADDA. Journal of Aerospace Power, 1 – 17 [2025-03-03]. https://doi.org/10.13224/j.cnki.jasp.20240346

[15]. (In Chinese)