Journal of Research in Science and Engineering (JRSE)
ISSN: 1656-1996 Volume-8, Issue-5, May 2026
DOI: 10.53469/jrse.2026.08(05).03

Dynamic Resource Scheduling Optimization under Industrial Internet-Enabled Smart Manufacturing Based on Deep Learning

Jiaqi Zhou

Business School, University of Shanghai for Science and Technology, Shanghai 200093, China
232481151@st.usst.edu.cn

Abstract: ***This study proposes a dynamic resource scheduling optimization method that integrates deep learning prediction to address the evolution and cooperative scheduling challenges of multiple types of dynamic resources in industrial internet-enabled manufacturing environments. First, a multidimensional input structure combining plan, state, and key performance indicator features is designed, and a Convolutional Neural Networks-Long Short-Term Memory-Attention based resource state prediction model is developed to jointly predict the future availability of two representative dynamic resources: machine and manpower. Second, based on the prediction results, a multi-objective scheduling optimization model is established with goals such as minimizing task completion time and maximizing resource utilization, and an improved Non-dominated Sorting Genetic Algorithm II is employed to achieve collaborative scheduling among multiple tasks and dynamic resources. Finally, simulation experiments are conducted in a representative intelligent manufacturing scenario to compare static, dynamic, and joint scheduling strategies. The results demonstrate that the proposed deep-learning-driven dynamic resource scheduling method effectively improves resource utilization, reducing completion time and task delays, thereby enhancing the responsiveness, and coordination of resource scheduling in industrial internet manufacturing systems. Furthermore, the obtained results provide a verifiable modeling and optimization framework for multi-resource collaborative scheduling in intelligent manufacturing.***

Keywords: Industrial Internet, Resource Prediction, Deep Learning, Dynamic Resource Scheduling, Multi-objective Optimization.

1. Introduction

Through rapid advancements in next-generation information technologies, the industrial internet is reshaping resource allocation and scheduling modes in manufacturing through its ubiquitous connectivity, data-driven intelligence, and open interoperability. By eliminating information silos across hierarchical and organizational boundaries, the manufacturing system has evolved from a closed and static structure to an open and dynamic one. Consequently, resource scheduling has shifted toward optimization processes oriented to dynamic resource characteristics, with management approaches becoming increasingly platform-based, service-oriented, and intelligent. Against this background, resource scheduling is no longer confined within a single enterprise, instead evolving into a collaborative process spanning multiple regions, entities, and systems. It has transformed from a "single-point optimization" problem into a systemic one that depends on state perception, information fusion, and dynamic evolution. However, in terms of practical production, frequent issues such as delayed task responses, unbalanced resource allocation, and high structural heterogeneity severely restrict task execution efficiency and resource coordination capability in multi-resource and multi-task scenarios [1]. As manufacturing systems evolve from "single-point scheduling" toward "cross-level collaboration," traditional static configuration modes can no longer meet the requirements of real-time scheduling and flexible response, resulting in dynamic resource scheduling gradually becoming a key research direction in industrial internet-enabled intelligent manufacturing environments [2]. He [3] extended dynamic scheduling research into the context of cloud manufacturing, constructing an interconnected service architecture that embodies the concept of industrial-internet-driven dynamic resource scheduling. Compared with static scheduling, dynamic resource scheduling emphasizes the perception and modeling of resource state variations, adjusting the scheduling plan cyclically according to resource availability during execution. In typical intelligent manufacturing scenarios, machine and manpower represent the two most prominent types of dynamic resources. The state of machine [4] is constrained by factors such as load rate, idle time, and inter-process dependencies, exhibiting strong temporal dynamics. Manpower resources [5] are highly uncertain owing to variations in worker skills and dynamic shift schedules, which significantly increase scheduling complexity. Therefore, developing dynamic perception and temporal prediction mechanisms for these two resource types and integrating them with task constraints for real-time matching and scheduling optimization provides a feasible approach to enhance the resource utilization, scheduling responsiveness, and operational flexibility of manufacturing systems under complex and dynamic conditions.

In recent years, various methods have been proposed for dynamic resource scheduling optimization. One focuses on resource optimization under computational support and edge–cloud collaboration [6]. Zhao [7] developed a cooperative tracking framework for Internet of Things (IoT) edge computing to improve resource allocation efficiency, Siatras [8] integrated digital twins with multi-agent systems to enhance the adaptability of production systems in multi-source resource collaboration and dynamic environments, and Zhang [9] combined distributed intelligence with dynamic resource configuration to achieve flexible on-demand responses. Cimino [10] proposed a simulation-based digital twin platform for internal supply chain management, integrating real-time data, predictive analytics, and scenario simulation to enhance coordination and adaptability across multi-plant manufacturing systems.

Another method focuses on perception mechanisms and multi-source information fusion. Sallam [11] proposed a scheduling method based on ensemble learning and data-driven perception to model dynamic resource constraints and improve scheduling accuracy and stability in dynamic environments. Similarly, Zhang [12] designed a multi-resource joint scheduling model to address flexible scheduling requirements across multiple manufacturing scenarios. However, although these approaches have benefited single-task scheduling and static resource configuration, most have not adequately considered the coupling mechanisms between resource state evolution and task scheduling, thus limiting their applicability in highly dynamic environments. To address this gap, Soltan and Ashrafi [13] proposed a model for multi-project environments by introducing a resource allocation performance index to enable real-time resource reconfiguration. Li [14] comprehensively considered multiple resource states such as equipment availability, and enhances scheduling adaptability in dynamic environments through a state – algorithm decoupling mechanism. Similarly, Barak [15] developed a scheduling model that accounts for sequence-dependent setup times under simultaneous machine and manpower constraints in dynamic resource environments, thereby illustrating an optimization pathway for multi-resource collaborative scheduling. Although these studies collectively advance the transition of resource scheduling from static matching to dynamic decision-making, most have overlooked the intrinsic dynamic evolution of resource states, failing to establish prediction-driven scheduling mechanisms that explicitly account for future resource availability.

In response, recent studies have begun introducing prediction mechanisms for resource state evolution. Deep learning methods not only autonomously learn scheduling rules and generate process allocation strategies—demonstrating stronger dynamic decision-making capabilities than do traditional optimization methods—but also model the temporal evolution characteristics of resources, thereby improving the foresight and responsiveness of scheduling systems regarding future resource availability [16]. Smendowski and Nawrocki [17] integrated multi-time-series forecasting for resource trend analysis and uncertainty management, enhancing the reliability of resource utilization prediction. Morariu [18] combined machine and deep learning models in large-scale manufacturing systems to jointly predict and optimize production planning, resource allocation, and scheduling, verifying that prediction-driven scheduling significantly improves makespan and resource utilization. Some existing studies still rely on traditional machine learning approaches, with research leveraging deep learning for resource prediction remaining limited and mostly confined to single-resource state prediction. Thus, models capable of capturing the cooperative evolution of multiple dynamic resources are lacking, restricting their applicability in multi-resource, multi-task scheduling scenarios under industrial internet environments.

Therefore, this paper focuses on two representative types of dynamic resources—machine and manpower—in industrial internet environments, conducting deep-learning-based temporal prediction of their state evolution. It described the development of a multi-objective dynamic resource joint scheduling optimization model integrating predictive perception, task execution efficiency, and resource utilization improvement. As illustrated in Figure 1, the overall research framework forms a dynamic closed loop composed of resource state feedback, production scheduling, and configuration optimization, to meet the high flexibility, responsiveness, and collaboration requirements of industrial internet manufacturing systems.

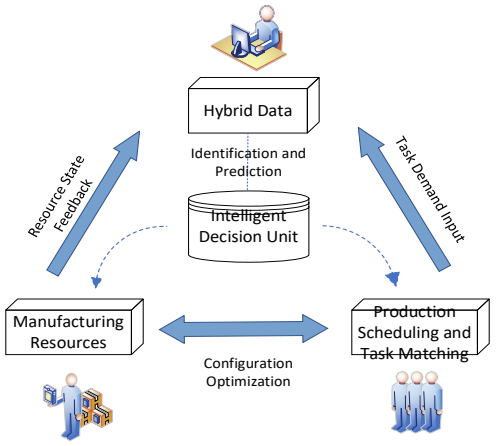


Figure 1: Flowchart of Resource State Identification and Scheduling Optimization.

2. Prediction of Resource States

In industrial internet environments, the dynamic and uncertain nature of resources is increasingly intensified, making accurate resource state prediction—a temporal modeling problem—a prerequisite for intelligent scheduling. Accordingly, the resource prediction problem in manufacturing scenarios is defined as using a given prediction horizon and historical resource input data to predict the resource usage state over multiple future time steps and estimate its availability.

2.1 Design of Multidimensional Input Features

To achieve high-precision prediction of future resource states, a comprehensive input feature system is constructed as the foundation for the deep learning prediction model. It integrates multi-source information such as task plans, resource states, and historical production data, forming a multidimensional input structure that encompasses both static constraints and dynamic evolution characteristics. This structure enables the model to capture temporal variations in resource states and cross-level correlations effectively. Its design follows the classical “Plan–State–Feedback” modeling logic in intelligent manufacturing systems, emphasizing the integration of task requirements (plan layer), resource state perception (state layer), and historical production data (key performance indicator [KPI] layer) to achieve predictive control and dynamic response capabilities [19]. Based on this framework, the input variables involved in resource availability prediction are categorized into the following three types of features:

(1) Planning: Representing the planned information of the current task and the expected resource input, reflecting the proactive objectives of resource scheduling.

(2) State: Describing the current availability of system resources, reflecting controllability and operational status.

(3) KPI: Reflecting the historical scheduling performance and system operational load of resources, introducing temporal feedback signals to improve prediction accuracy.

Seven key input variables are designed, covering three planning, two state, and two KPI features, as shown in Table 1.

Table 1: Multi-dimensional Input Feature Table

No.	Feature Name	Type	Description
1	Process time	Planning feature	Processing time required for a single operation
2	Available machine duration	State feature	Remaining operational time available for the machine
3	Available personnel	State feature	Number of workers currently available for assignment to this process
4	Personnel requirement	Planning feature	Minimum manpower requirement for the process
5	Remaining delivery time	Countdown feature	Remaining time until the delivery deadline
6	Historical machine load rate (%)	KPI feature	Average utilization intensity of machine over a historical period
7	Historical manpower tension (0-1)	KPI feature	Degree of manpower shortage observed during the historical period

These features encompass key resource dimensions in intelligent manufacturing, accounting for both horizontal resource supply-demand relationships and vertical historical evolution trends. To meet the structural requirements of sequential input for the deep learning model, all features are organized in a time-series format to construct the input tensor $X \in \mathbb{R}^{T\times D}$, where T denotes the time-window length (number of past time steps observed during each training or prediction iteration) and D represents the feature dimension (number of variables at each time step).

2.2 Development of the Prediction Model

Traditional resource prediction methods often fail to capture the complex spatiotemporal dynamics and long-term dependencies inherent in production processes, making resource state prediction particularly challenging in cross-resource manufacturing scenarios with uncertain or rapidly changing demands. To simultaneously extract local temporal features and model global state evolution, this study proposes a hybrid deep-learning-based multi-resource state joint prediction model that integrates Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), an attention mechanism — CNN-LSTM-Attention — and a multi-task learning (MTL) decoupling output module. The overall framework of the model is illustrated in Figure 2.

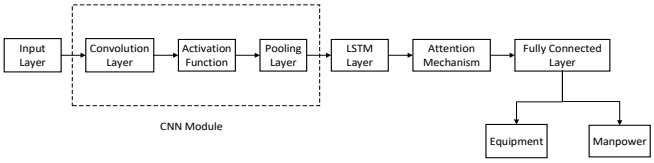


Figure 2: Framework of the Convolutional Neural Networks – Long Short-Term Memory (CNN-LSTM) Attention Model.

(1) Feature Extraction Layer (CNN)

A one-dimensional convolutional network is employed to extract local features from the multidimensional time-series input, identifying short-term variation patterns of variables, such as process time, available machine duration, and available personnel. Each convolutional layer is followed by a Rectified Linear Unit (ReLU) activation function to introduce nonlinearity and a max-pooling layer to compress features and enhance temporal shift invariance.

(2) Temporal Modeling Layer (LSTM)

The temporal embedding vectors generated by the CNN are fed into a multi-layer LSTM network to capture long-term evolution patterns of manufacturing resource states, such as remaining delivery time and historical machine load rate. This enhances the model's ability to memorize and represent resource evolution trends over extended horizons.

(3) Attention Mechanism Layer (Attention)

To mitigate information dilution in long-sequence modeling and enhance the model's ability to identify key temporal features and critical state transitions, an attention mechanism is introduced atop the CNN-LSTM multi-task prediction structure. This automatically learns the importance of each time step and focuses on the most representative temporal segments for the current prediction task. Its computation is as follows:

$$e^t = v^T tanh(Wh_t + b) \quad (1)$$

$$\alpha_t = \frac{exp(e_t)}{\sum_{k=1}^{T} exp(e_k)} \quad (2)$$

$$c = \sum_{t=1}^{T} \alpha_t h_t \quad (3)$$

Specifically, Eq.(1) is a scoring function whose scalar output e^t denotes the attention score at time step t. Here, W is a trainable weight matrix that maps h_t to the attention space, b is a bias term, tanh(·) is a nonlinear activation, and v^T is another trainable vector. Because e^t is a score rather than a weight, it must be normalized: Eq.(2) applies a softmax to obtain weights $\alpha_t \in [0,1]$, which quantify the degree of attention assigned to the t-th time step. Finally, Eq.(3) computes the context vector $c \in \mathbb{R}^d$ as the weighted sum of the hidden states, representing a compact summary of the most informative content in the entire sequence for the current prediction task.

(4) Multi-Task Decoupling Layer (MTL)

Through "shared-decoupled" architecture, the model performs joint learning across all resources while retaining the ability to learn resource-specific mapping relationships. Multiple task-specific output heads in the decoupling layer correspond to different resource categories, each implemented via an independent, fully connected network. These output heads process the shared temporal features to predict the state of each resource type within the future time window.

2.3 Prediction–Scheduling Coupling Mechanism

To integrate deep learning with scheduling optimization, a coupling mechanism is established between the prediction and scheduling models. The prediction module estimates the availability of machine and manpower over a future time window based on historical features and outputs a resource

availability matrix that is converted into explicit "available or unavailable" labels and fed directly into the scheduling model as feasibility constraints after threshold-based binarization. Building on this, the scheduling model allocates tasks within feasible intervals via a sliding-window mechanism, achieving real-time alignment between task assignment and the predicted availability. This coupling ensures that deep learning prediction is not an isolated component but a driving input to the scheduling optimization.

3. Dynamic Resource Scheduling Optimization Model

3.1 Problem Description

Because each production unit has a limited resource pool, different resources across various units may differ in processing efficiency, cost, and duration for the same task. Therefore, resource allocation strategies should be dynamically determined based on predicted resource states and task requirements. Unlike traditional static scheduling models, the proposed optimization model incorporates not only the current state of resources but also predicted future availability constraints. By integrating prediction and scheduling, the scheduling scheme dynamically adjusts with time, reflecting changes in resource availability across future time intervals and, thus, modeling and exploiting system dynamics. Given a set of tasks and resource constraints, the scheduling optimization model leverages the predicted resource availability data to simultaneously optimize five objectives—task completion efficiency, delay minimization, resource usage cost, resource utilization, and demand satisfaction—achieving coordinated scheduling and configuration optimization across multiple manufacturing units and tasks.

Let us suppose that a customer order contains a set of part-processing tasks $T = \{T_1, T_2, \dots, T_n\}$, where n denotes the total number of tasks, and each part consists of s_i operations, and the set of available manufacturing resources is denoted as $R = \{R_m, R_l, R_k\}$. To capture the temporal variation of resource availability, the CNN-LSTM-Attention prediction model generates resource availability probabilities, which are binarized based on a predefined threshold $A_i(t)$. These binary predictions over the future time window serve as input variables to the dynamic resource scheduling model.

3.2 Notations

Table 2: Model Symbol Explanation

Notation	Definition
$R_m = \{1, \dots, i\}$	Total number of available machines
$R_l = \{1, \dots, \mathrm{k}\}$	Total number of available personnel
$\mathrm{T} = \{\mathrm{t}_1, \dots, \mathrm{t}_j\}$	Set of tasks to be scheduled
$\mathrm{x_{ij}} \in \{0,1\}$	Binary variable: 1 if machine i is assigned to task j, otherwise 0
$h_{\mathrm{jk}} \in \{0,1\}$	Binary variable: 1 if personnel k is assigned to task j, otherwise 0
$\mathrm{D_j}$	Total capacity required by task j
H_{j}	Manpower required by task j
A_{i}	Maximum capacity of machine i
P_{ij}	Capacity contribution of machine i to task j
S_k	Availability status of personnel k (1 if free, 0 if occupied)
d_j	Delivery deadline of task j
t_j^{end}	Completion time of task j

3.3 Model Formulation

For simplicity, the following assumptions are made:

The resource types are functionally exclusive, with no cross-type substitution or hybrid usage;

All tasks are known in advance, with clear total demand;

A resource can, at any time, process at most one operation of one part;

Each operation of a part can only be processed by one resource at any time;

The model considers only sequential manufacturing structures.

3.3.1 Objective Functions

The scheduling optimization model aims to minimize task completion time, delayed task count, resource configuration cost, and total resource occupancy, while maximizing overall resource utilization.

Task Completion Time:

$$f_1 = \sum_{j=1}^{n} t_j^{end} \tag{4}$$

Number of Delayed Tasks:

$$f_2 = \sum_{j=1}^{n} \prod\left(t_j^{end} > d_j\right) \tag{5}$$

Resource Configuration Cost:

$$f_3 = \sum_{i,j} c_{ij}^{(d)} \cdot x_{ij} + \sum_{j,k} c_{jk}^{(h)} \cdot h_{jk} \tag{6}$$

Total Resource Occupancy:

$$f_4 = \sum_{i,j} x_{ij} + \sum_{j,k} h_{jk} \tag{7}$$

Resource Utilization:

$$f_5 = \sum_{i=1}^{m}\left(A_i - \sum_{j=1}^{n} x_{ij} \cdot P_{ij}\right) + \sum_{k=1}^{l}\left(S_k - \sum_{j=1}^{n} h_{jk}\right) \tag{8}$$

Combined Objective Function:

$$min(Z_1) = f_1 + f_2 + f_3 + f_4 + f_5 \tag{9}$$

Among these, Eq.(6) represents the total allocation cost of the two types of resources, whereas Eq.(8) aims to maximize the ratio of resources actually utilized, thereby reducing idle resources.

3.3.2 Constraints

Demand Satisfaction Constraint:

$$\sum_{i=1}^{m} x_{ij} \cdot P_{ij} \geq D_j, \sum_{k=1}^{l} h_{jk} \geq H_j\,, \forall j, q \tag{10}$$

Resource Capacity Constraint:

$$\sum_{j=1}^{n} x_{ij} \cdot P_{ij} \leq A_i, \sum_{j=1}^{n} h_{jk} \leq S_k\,, \forall i, k, q \tag{11}$$

Eq.(10) represents the demand satisfaction constraint, which ensures that the required machine capacity and manpower for each task are fully met. Eq.(11) represents the resource supply constraint, ensuring that the total allocated amount of a given resource across all tasks does not exceed its predicted

availability over the entire planning horizon. Unlike traditional scheduling models that assume fixed resource capacities, both A_i and S_k are time-dependent variables that are dynamically updated according to the prediction results.

3.3.3 Dynamic Resource State Constraint

Traditional scheduling models typically assume that resources remain available throughout the planning period; however, this assumption is inconsistent with the dynamically changing characteristics of resources in industrial internet manufacturing environments. To enhance the model's adaptability to real-world production conditions, dynamic resource state constraints are introduced based on predictive availability:

$$\begin{cases} x_{ij} = 0, \text{if machine i is inactive} \\ h_{jk} = 0, \text{if worker k is unavailable} \end{cases} \quad (12)$$

Here, x_{ij} and h_{jk} depend on the results of the predicted availability matrix, reflecting the dynamic temporal evolution of resource availability and ensuring consistency between task allocation and actual resource states. The core idea is that by introducing predictive constraints on future resource states, the scheduling scheme can fully reflect resource dynamics, avoiding unrealistic situations in static models wherein unavailable resources are forcibly allocated. This approach significantly enhances the adaptability and robustness of the scheduling results to resource fluctuations in industrial internet manufacturing environments. This constraint establishes an explicit coupling between the prediction and scheduling models, enabling the scheduling process to dynamically respond to temporal variations in resource states and, thus, resource availability over time.

4. Improved Non-dominated Sorting Genetic Algorithm II (NSGA-II)

To efficiently solve the prediction-driven dynamic resource scheduling optimization model established in Section 3, this study improves upon the traditional Non-dominated Sorting Genetic Algorithm II (NSGA-II). To address issues such as premature convergence and uneven distribution of solutions, the algorithm breaks the limitations of one-dimensional optimization and incorporates dynamic constraint conditions into the search process. An improved NSGA-II that integrates dynamic crowding-distance evaluation, stage-switch sorting, adaptive crossover and mutation control, and an elite-preservation mechanism is designed to effectively handle the complex solution space under prediction-driven constraints, ensuring that the generated scheduling schemes conform to the temporal evolution characteristics of future resource states.

4.1 Dynamic Crowding-Distance Calculation Driven by Local Neighborhood Density

A local neighborhood-density factor is introduced as a correction term to the traditional crowding distance calculation, allowing the crowding distance to adjust dynamically with population distribution and thereby improve the uniformity of solutions along the Pareto front. Specifically, for any individual x_i in the population, the neighborhood set in the objective space is defined as:

$$N(x_i) = \{x_j | d(x_i, x_j) \le \epsilon, j \ne i\} \quad (13)$$

Where $d(x_i, x_j)$ denotes the Euclidean distance between individuals x_i and x_j in the objective space, and ϵ is the neighborhood radius. The neighborhood density is then calculated as:

$$\rho(x_i) = \frac{1}{1+\frac{1}{|N(x_i)|}\sum_{x_j \in N(x_i)} d(x_i, x_j)} \quad (14)$$

This value reflects how crowded the region around an individual is; the neighborhood-density factor is then incorporated into the conventional crowding distance computation. If $D(x_i)$ denotes the original crowding distance of individual x_i, the corrected dynamic crowding distance is defined as:

$$D'(x_i) = \frac{D(x_i)}{1+\lambda \cdot \rho(x_i)} \quad (15)$$

Here, λ is a tuning parameter that controls the influence of neighborhood density on crowding distance. By reducing the crowding distance value of individuals in highly dense regions, the algorithm prevents excessive local clustering and enhances the global uniformity of the Pareto solution set.

4.2 Stage-Based Sorting-Strategy Switching

In the standard NSGA-II, individual selection primarily relies on fast, non-dominated sorting and crowding distance comparison. However, this fixed strategy cannot balance global exploration and local convergence effectively across different evolutionary stages. Therefore, the algorithm adopts fast, non-dominated sorting during early iterations to expand the coverage of the Pareto front, ensuring global exploration. As population diversity decreases in later stages, it switches to a boundary-solution-prioritized sorting strategy, accelerating local convergence and improving the stability of frontier solutions.

4.3 Adaptive Control of Crossover and Mutation Probabilities

In traditional genetic algorithms, crossover and mutation probabilities (P_c and P_m) are fixed, making it difficult to balance exploration and exploitation across different evolutionary stages. Thus, a convergence-trend-driven adaptive adjustment strategy is introduced to dynamically balance the algorithm's global exploration and local exploitation capabilities. The core idea is that when the population converges too quickly, P_m is increased to enhance perturbation and avoid premature convergence, whereas when the solution distribution becomes sparse, P_c is increased to strengthen global search and coverage. The adaptive adjustment formulas are as follows:

$$P_c(t) = P_c^{min} + (P_c^{max} - P_c^{min}) \cdot \left(1 - \frac{D_t}{D_0}\right) \quad (16)$$

$$P_m(t) = P_m^{max} - (P_m^{max} - P_m^{min}) \cdot \left(1 - \frac{D_t}{D_0}\right) \quad (17)$$

Where D_t denotes the diversity metric of generation t and D_0 is the initial diversity level. This strategy adaptively adjusts genetic operation parameters based on convergence trends to maintain active exploration throughout evolution.

4.4 Elite Preservation Mechanism

The proposed elite-preservation mechanism incorporates a fitness-layering concept into the non-dominated sorting framework, ensuring the stable inheritance of high-quality solutions while maintaining population diversity. First, the population is divided into multiple layers based on fitness levels. Then, combining crowding distance thresholds with diversity metrics, the elite proportion in each layer is dynamically adjusted to prevent excessive concentration of individuals in specific regions. By aggregating elites across all layers, a representative and evenly distributed subset is obtained. The elite ratio is calculated as follows:

$$\alpha_l = \alpha_0 \cdot \frac{1}{1+\lambda\rho_l} \tag{18}$$

Where α_l denotes the elite ratio of layer l, ρ_l is the average neighborhood density of that layer, λ is a tuning coefficient, and α_0 represents the initial elite proportion.

The overall workflow of the improved algorithm is illustrated in Figure 3.

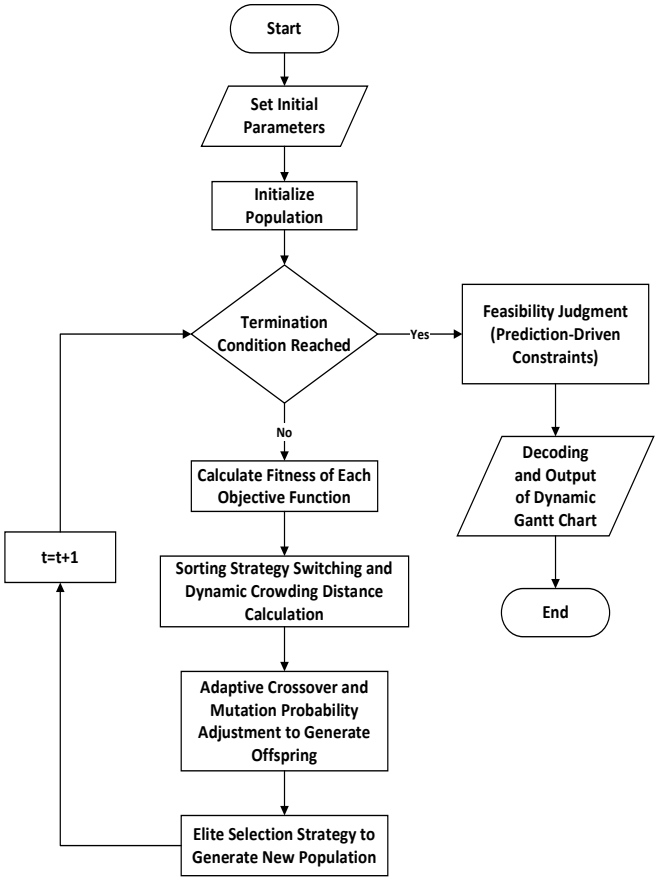


Figure 3: Flowchart of the Improved Non-dominated Sorting Genetic Algorithm II (NSGA-II).

5. Experiments and Results Analysis

5.1 Experimental Analysis of the Prediction Model

The effectiveness of the proposed CNN-LSTM-Attention model for resource-state prediction is assessed from two perspectives: the training process and comparative prediction against representative time-series models. The experiments are conducted in a Python 3.11.4 environment using 1,000 historical records of manufacturing resource states, featuring fields defined in Table 1. The loss curves during training and validation are shown below.

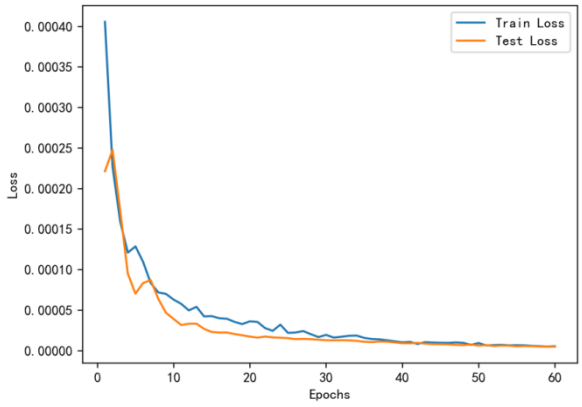


Figure 4: Validation Loss Curve.

The overall curves in Figure 4 reveal that the loss drops rapidly within the first 10 epochs, indicating fast convergence during training and suggesting that the model quickly captures the key relationships between input features and target variables, completing the initial parameter tuning and feature fitting. Throughout the process, training and validation losses follow consistent trajectories with a small gap and no rebound on the validation set, indicating no evident overfitting and sound generalization capability.

For benchmarking, two canonical recurrent models—LSTM and Gated Recurrent Unit [GRU]—are selected and compared with the proposed model on the same dataset using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as evaluation metrics.

$$E_{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\widehat{y_i} - y_i| \tag{19}$$

$$E_{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{20}$$

Here, y_i denotes the ground truth and $\hat{y}_i$ denotes the prediction. MAE measures the absolute deviation between predicted and actual resource states, while RMSE is more sensitive to large deviations.

As shown in Tables 3 and 4, the proposed CNN-LSTM-Attention model significantly outperforms LSTM and GRU, achieving the lowest errors on both MAE and RMSE, indicating stronger robustness and generalization. Notably, its RMSE fluctuates least, revealing more stable fits and accurate forecasts of resource availability over future windows. In comparison, although the LSTM and GRU models demonstrate predictive capability on specific tasks, their overall error levels remain significantly higher than those of the CNN-LSTM-Attention model, indicating inherent limitations in capturing temporal dependencies and key feature representations. Therefore, the above evaluation metrics reveal that the CNN-LSTM-Attention model is more suitable for multi-task resource-state prediction scenarios of this type.

Table 3: MAE of the Three Models on the Resource Prediction Task

MAE	LSTM	GRU	CNN-LSTM-Attention
machine load rate (%)	0.036	0.151	0.010
Manpower tension	0.041	0.111	0.017

Table 4: RMSE of the Three Models on the Resource Prediction Task

RMSE	LSTM	GRU	CNN-LSTM-Attention
machine load rate (%)	0.08	0.21	0.02
Manpower tension	0.06	0.20	0.03

5.2 Threshold Setting

Neural network models typically output continuous probability values that only indicate the likelihood of future resource availability and cannot be directly consumed by the scheduler. To enable efficient use of deep-learning predictions in dynamic resource scheduling, these continuous values must be discretized into explicit decision signals — "schedulable" or "unschedulable." The threshold calibration framework proposed by Sahoo at NeurIPS indicates that mapping predicted probabilities to threshold-based binary decisions is crucial to improve the decisional utility of predictions, particularly for threshold-driven tasks such as scheduling and resource allocation. Inspired by this line of research, a threshold-based binarization strategy is employed to convert the predicted probabilities of dynamic resource availability into label signals executable by the scheduler. To introduce a threshold parameter $\theta \in [0,1]$, the mapping is given as follows:

$$AF_t = \begin{cases} 1, \hat{p_t} \geq \theta \\ 0, \hat{p_t} < \theta \end{cases} \quad (21)$$

When the predicted probability is no less than the threshold, the resource is labeled 1 (schedulable); otherwise 0 (unschedulable). These labels serve as feasibility filters and dynamic decision logic within the scheduler.

Although conventional approaches often employ a unified threshold setting for rapid deployment, the response differences of various resource types to predicted probabilities is neglected, leading to blurred decision boundaries and consequently affecting resource utilization efficiency and timely task completion. To address this problem, the threshold selection method based on Economic Value (EV) proposed by Wu [20] is adopted. By constructing EV curves under different thresholds, the positive contribution of prediction results to actual scheduling performance is evaluated to determine the optimal threshold that maximizes scheduling benefits. The EV metric, originally derived from cost–loss analysis in meteorology, is redefined in this study as a comprehensive measure that integrates scheduling costs, penalties for scheduling failures, and task delay losses, thereby quantifying the decision-support capability of predicted resource availability probabilities under varying threshold settings. By formulating an EV function that jointly considers "misjudgment costs in resource scheduling" and "costs of resource waste, separate EV-threshold curves are constructed for machine and manpower resources. The optimal thresholds therein derived serve as critical parameter references for resource-scheduling strategies driven by predictive results.

According to the results shown in Figure 5, system benefits are maximized when the machine-resource threshold is $\theta = 0.70$ and the manpower-resource threshold is $\theta = 0.65$. Accordingly, the optimal thresholds for machines and manpower are respectively adopted for binarization of predicted resource availability.

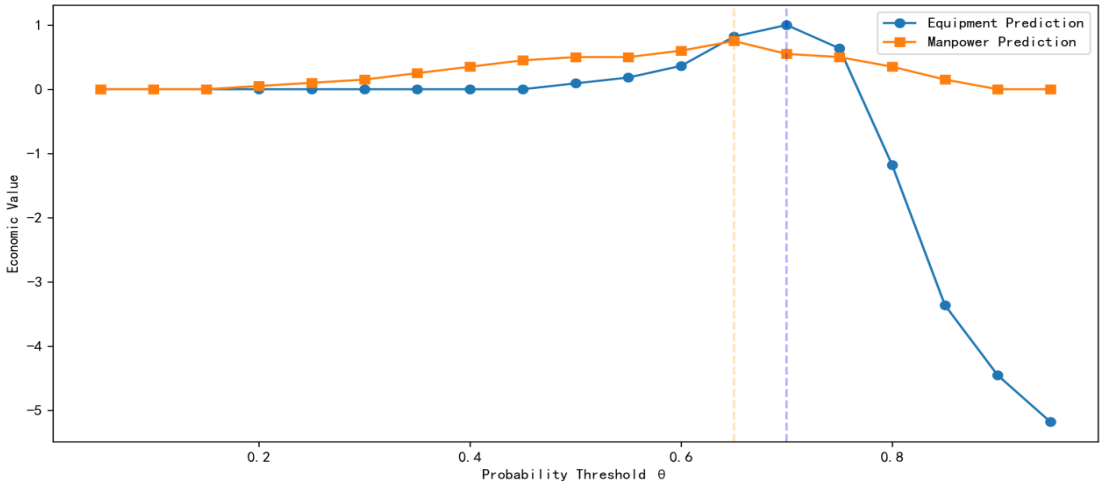


Figure 5: EV – Threshold Curves

5.3 Simulation

To validate feasibility and effectiveness in an industrial internet context, we simulate a smart workshop for an automotive engine control module: 60 tasks to be scheduled, 9 ongoing task allocations (fixed), resource availability provided by the predictor, 12 machines and 15 workers available. Three strategies are compared: static, dynamic, and joint dynamic scheduling. Static scheduling performs one-shot allocation based on current states. Dynamic scheduling adds single-resource predictive availability constraints. Joint dynamic scheduling simultaneously considers predicted availability of both machines and manpower within a unified optimization framework to emulate multi-resource parallel constraints in practice. Table 5 lists the main algorithmic settings used in the simulation.

Table 5: Parameter Settings

Variable Name	Assigned Value
population_size	300
iteration	50
crossover_rate	[0.28,0.88]
variation_rate	[0.05,0.20]
elite_ratio	5%

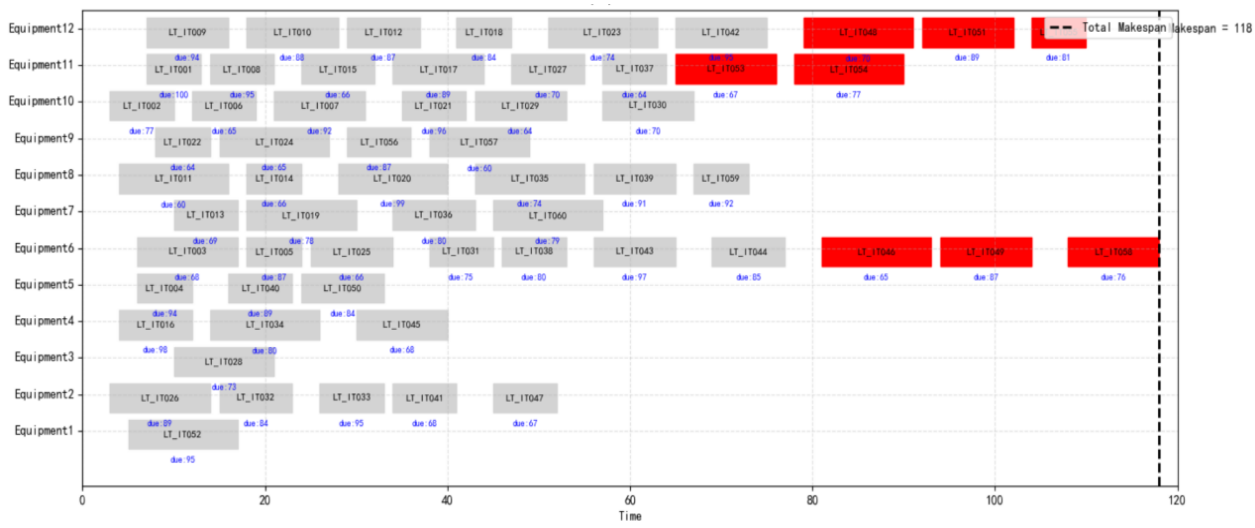
The simulation dataset is shown in Table 6 (10 records displayed).

Table 6: Simulation data (excerpt)

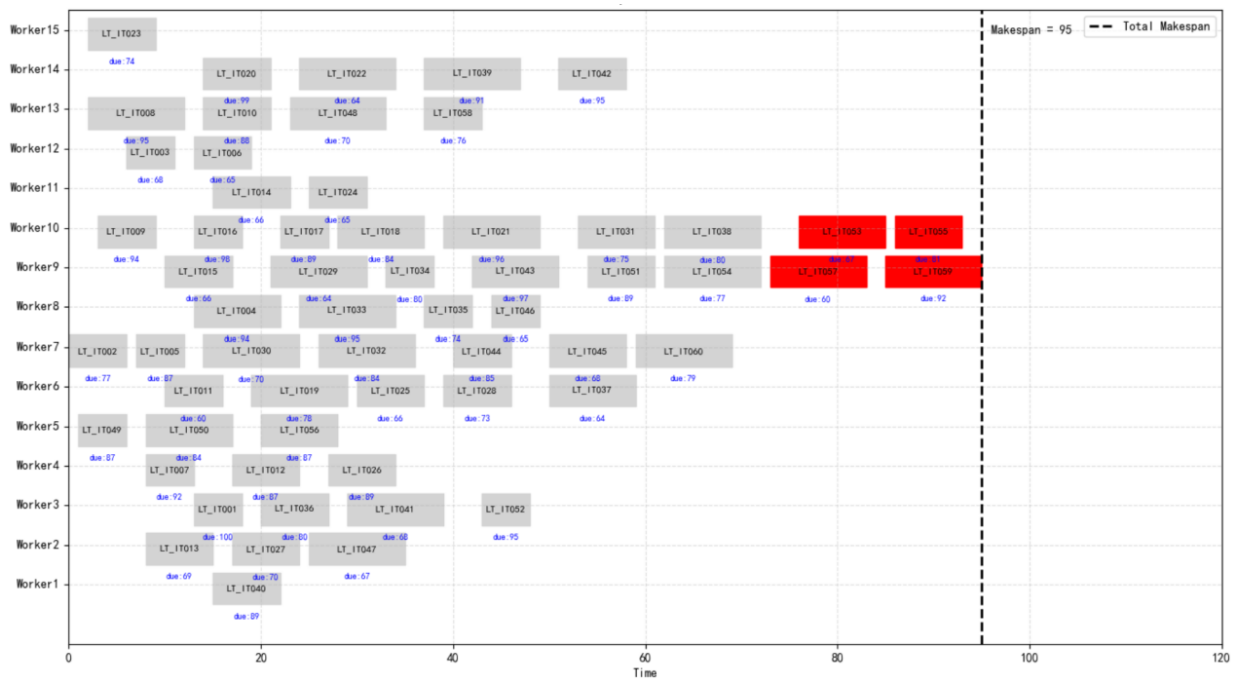
Task ID	Delivery Time (h)	Production Duration (h)	Machine Availability	Manpower Availability	Task Urgency Level
LT_IT030	70	7	[0,1,1,0,1,0,1,1,0,0,0,1]	[0,1,1,0,0,1,1,0,0,0,0,0,1,1,1]	High Urgency
LT_IT031	75	7	[0,1,1,0,0,1,1,0,0,1,0,0]	[0,1,1,1,1,1,1,1,0,0,0,0,0,0,0]	High Urgency
LT_IT032	84	6	[0,1,0,0,1,0,1,0,1,1,1,0]	[1,0,0,0,0,1,1,0,0,1,0,1,0,0,1]	Medium Urgency
LT_IT033	95	7	[0,0,0,1,0,0,0,0,1,1,0,1]	[0,0,0,1,1,1,1,1,1,1,1,1,0,1,1]	Low Urgency
LT_IT034	80	7	[1,1,1,1,1,1,0,0,1,1,0,1]	[0,0,0,0,0,1,0,1,1,0,0,1,0,0,1]	High Urgency
LT_IT035	74	6	[0,1,1,1,1,1,0,0,0,0,1,0]	[1,0,1,1,0,0,0,0,1,0,0,1,0,0,0]	Medium Urgency
LT_IT036	80	7	[0,0,0,0,0,1,0,1,1,0,0,0]	[0,1,0,0,0,0,0,0,1,1,0,0,0,1,0]	High Urgency
LT_IT037	64	6	[0,0,1,1,0,0,1,1,1,1,0,1]	[0,1,1,1,1,1,0,0,1,1,0,1,1,0,1]	High Urgency
LT_IT038	80	6	[1,0,0,0,0,0,0,1,1,0,0,0]	[1,1,1,1,1,1,1,1,1,0,0,0,0,1,1]	Medium Urgency
LT_IT039	91	8	[1,0,1,1,0,1,0,0,0,0,1,1]	[1,0,1,0,1,1,1,0,1,1,0,1,0,1,0]	Medium Urgency
LT_IT040	89	7	[0,0,0,0,0,1,0,0,1,1,1,1]	[1,1,1,0,0,1,0,0,0,1,1,0,1,0,0]	Medium Urgency

5.2.1 Analysis of Experimental Results

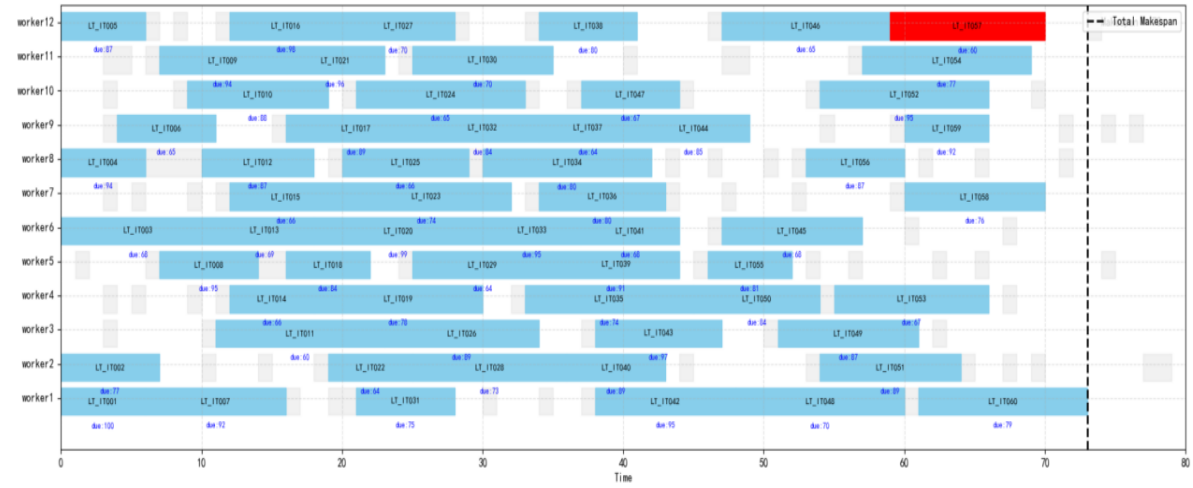
Without prediction, the static scheduler performs a one-shot allocation based on current states and cannot sense future changes. As shown in Figures 6 (a) and (b), pronounced resource stacking and temporal congestion occur: task blocks cluster along the machine and manpower axes and some resources remain idle while others are overloaded, yielding imbalanced utilization, extended makespan, and many delayed tasks.



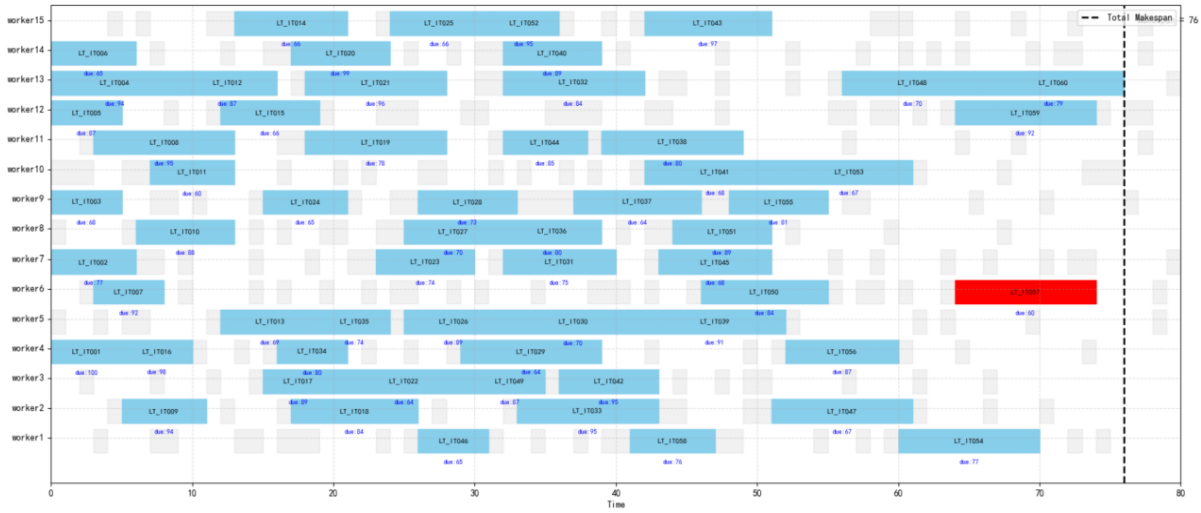
(a)



(b)



(c)



(d)

Figure 6: Static/Dynamic Mahine and Manpower Gantt Chart

With predictive availability constraints (Figures 6 (c) and (d)), tasks avoid predicted unavailable intervals and are advanced/deferred into feasible windows: the overall schedule length shrinks markedly, delays decrease, and utilization improves. The sliding-window allocation raises feasibility and mitigates late-stage backlog risks, showing good adaptability.

Figure 7 depicts the machine–manpower joint-cooperative schedule under predictive constraints: the model enforces synchronized temporal matching for both resource types via continuous feasible windows from the availability matrices. The makespan is further reduced with the lowest delay ratio, task blocks become highly parallel and coordinated, and both horizontal redundancy and vertical overlap are mitigated. Machines and workers demonstrate optimized cooperative occupation and interleaved utilization. These results indicate that the joint scheduling model—integrating predictive guidance with collaborative resource coordination—can minimize overall resource occupancy and maximize system efficiency without compromising task timeliness.

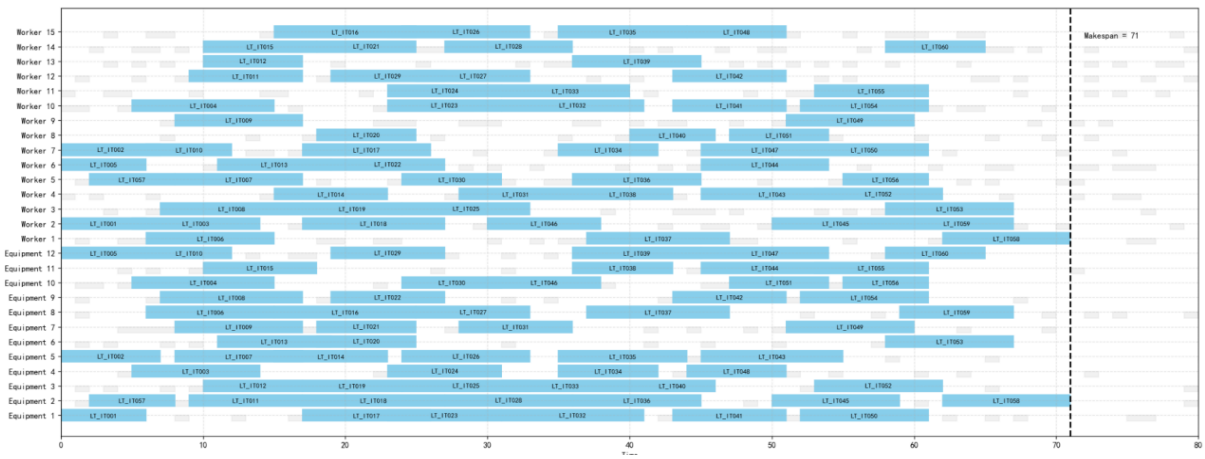


Figure 7: Joint Dynamic Resource Scheduling Gantt Chart

Table 7: Design Specification of Disturbance Levels

Level	Dynamic Disturbance Characteristics	Availability-Prediction Disturbance Characteristics	Scenario Description
Level 1	Small disturbance amplitude (±10%); errors approximately symmetric	Slight availability errors; no structural bias across resources （p01 ≈ p10）	Simulates mild noise or edge fluctuations affecting the scheduler; overall system response remains stable, with only occasional misjudgment of availability for local resources.
Level 2	Medium disturbance amplitude (±15%); errors asymmetric	Availability of certain key resources is underestimated; an unbalanced disturbance structure arises (e.g., p01 > p10)	Simulates amplified bias in predicting key resources, causing premature “unavailable” judgments on critical-path resources and revealing the scheduler’s sensitivity to misclassification
Level 3	Large disturbance amplitude (±20%-25%); errors strongly directional	Multiple resource types are simultaneously underestimated; key machines and workers are broadly constrained (p01 ≫ p10)	Simulates extreme resource-disturbance scenarios such as clustered machine failures or mass worker absence, reflecting the robustness boundary of the scheduling algorithm under extreme uncertainty

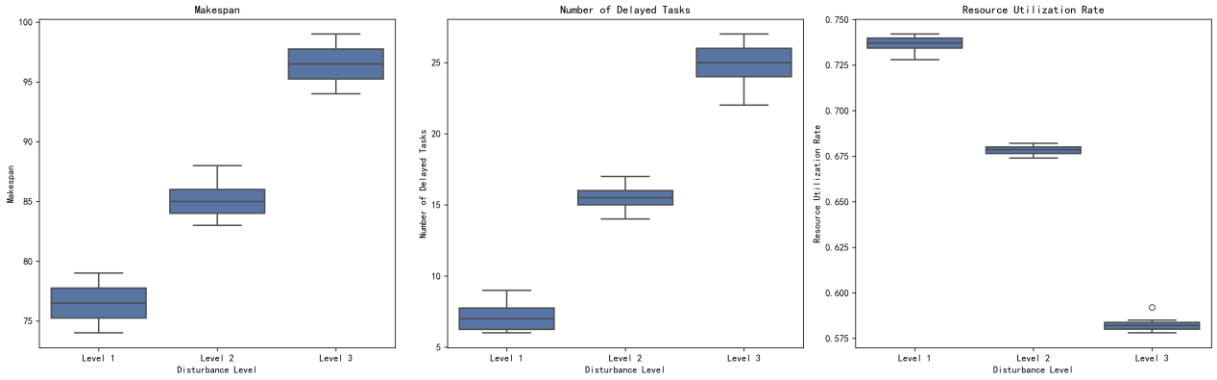


Figure 8: Boxplots of Scheduling Performance under Different Disturbance Levels

5.2.2 Sensitivity Analysis

To further evaluate the robustness of the dynamic resource scheduling model under prediction-error disturbances, three disturbance levels are constructed to simulate the dynamic evolution of prediction errors. They range from slight fluctuations to biased disturbances and then to severe miscalibration. These levels are mainly controlled by adjusting the ratio between p01 and p10 and the disturbance coverage rate, thereby controlling the intensity and direction of disturbance. The specific setups are shown in Table 7.

As shown in Figure 8, system performance degrades markedly with increasing disturbance levels. Regarding makespan, the median number of scheduling steps increases from approximately 76 at Level 1 to approximately 96 at Level 3, indicating increased task congestion and longer resource waiting times. The number of delayed tasks increases from 7 to approximately 26, and a clear clustering of delays emerges at Level 3, revealing high sensitivity to misjudgment of critical-path resources. Resource utilization drops from 0.735 to 0.595, reflecting substantial forced idleness caused by scheduling misclassification and a consequent limit on overall efficiency.

Overall, the scheduler exhibits good robustness at Level 1. Level 2 shows a clear diffusion of scheduling pressure, with task backlogs arising particularly when key resources are misclassified. Level 3 presents a distinct performance inflection: the scheduling logic fails under highly skewed error disturbances, some tasks face severe accumulation, and resource utilization plunges–characteristics of a typical robustness boundary. These observations indicate that although the availability-prediction-based dynamic scheduling model validated herein possesses strong fault tolerance under low-to-moderate disturbances, its robustness degrades rapidly when errors are highly asymmetric and extremely biased.

6. Conclusion

This paper addresses the optimization of multi-dynamic resource scheduling under industrial internet-enabled manufacturing, focusing on the deep integration mechanism between resource-state prediction and task scheduling. It proposes a resource joint scheduling method incorporating CNN-LSTM-Attention–based prediction. By constructing a multi-dimensional input feature system and a resource-state prediction model, temporal perception of the availability of machines and manpower resources is achieved. Furthermore, based on the prediction results, a multi-objective dynamic resource scheduling model is developed with objectives including minimum makespan, task delays, and allocation cost. The improved NSGA-II is employed to achieve efficient search within the solution space and elite preservation. Experimental results demonstrate that the proposed method exhibits strong adaptability and scheduling performance under complex resource constraints and dynamically changing conditions and verify that, in the joint scheduling of typical dynamic resources—machines and manpower—the prediction-driven mechanism further enhances coordination among multiple resources. However, this study primarily focuses on scheduling optimization for dynamic resources, in which forward-looking prediction of resource states improves scheduling performance, without incorporating real-time dynamic adjustment during scheduling execution. Future work should compare the prediction accuracy and scheduling performance under different prediction-model structures while maintaining the current optimization framework, aiming to enhance the robustness of the integrated prediction–scheduling system. Additionally, future research should explore the dual optimization of resource dynamics and scheduling dynamics by integrating prediction-driven mechanisms with real-time scheduling, thereby achieving full-chain intelligent optimization—from resource-state perception to scheduling-process adaptation.

Disclosure Statement

The authors have no relevant financial or nonfinancial interests to disclose.

Notes on Contributors

Jiaqi Zhou is a Master's student in Management Science and Engineering at the School of Management, University of Shanghai for Science and Technology. Her research interests include industrial interconnection and smart manufacturing. Email: 232481151@st.usst.edu.cn

Data availability statement

The data supporting the findings of this study are available from the corresponding author upon reasonable request

References

[1] Chao, Y., Zhuang, C., Guo, H., & Liu, J. (2026). A genetic programming hyper-heuristic with whale optimization algorithm for the dynamic resource-constrained multi-project scheduling problems. *Expert Systems With Applications, 295*, 295128881–295128881. https://doi.org/10.1016/j.eswa.2025.128881

[2] Khakifirooz, M., Fathi, M., & Dolgui, A. (2024). Theory of AI-driven scheduling (TAIS): A service-oriented scheduling framework by integrating theory of constraints and AI. International Journal of Production Research, 1–35. https://doi.org/10.1080/00207543.2024.2424976

[3] He, J., Wu, J., Ni, J. et al. (2025). Real-time task scheduling strategy for 3D printing cloud platforms in health scenes. Appl Intell 55, 1002. https://doi.org/10.1007/s10489-025-06907-2

[4] Lu, S., Pei, J., Liu, X., & Pardalos, P. M. (2021). A hybrid DBH-VNS for high-end equipment production scheduling with machine failures and preventive maintenance activities. Journal of Computational and Applied Mathematics, 384. https://doi.org/10.1016/j.cam.2020.113195

[5] Oladele, I. O., Okoro, C., Falana, S., Olanrewanju, O. F., Adelani, S. O., & Onuh, L. N. (2024). Advancement of human-machine collaboration in manufacturing: A review on industry 1.0–6.0(J). Journal of Mechanical Engineering and Sciences, 10303–10329. https://doi.org/10.15282/jmes.18.4.2024.7.0813

[6] Guo, L., He, Y., Wan, C., Li, Y., & Luo, L. (2024). From cloud manufacturing to cloud–edge collaborative manufacturing. Robotics and Computer-Integrated Manufacturing, 90, 90102790–. https://doi.org/10.1016/j.rcim.2024.102790

[7] Zhao, Z., Lin, P., Shen, L., Zhang, M., & Huang, G. Q. (2020). IoT edge computing-enabled collaborative tracking system for manufacturing resources in industrial park. Advanced Engineering Informatics, 43, 43101044–43101044. https://doi.org/10.1016/j.aei.2020.101044

[8] Siatras, V., Bakopoulos, E., Mavrothalassitis, P., Nikolakis, N., & Alexopoulos, K. (2024). Production scheduling based on a multi-agent system and digital twin: A bicycle industry case. Information, 15(6), 337. https://doi.org/10.3390/info15060337

[9] Zhang, T., & Wu, Y. (2024),220119714-. Collaborative allocation model and balanced interaction strategy of multi flexible resources in the new power system based on Stackelberg game theory. Renewable Energy, 0960-1481.https://doi.org/10.1016/j.renene.2023.119714

[10] Cimino, A., Longo, F. et al. (2024). Enhancing internal supply chain management in manufacturing through a simulation-based digital twin platform, Computers & Industrial Engineering, 198, 110670, https://doi.org/10.1016/j.cie.2024.110670

[11] Sallam, K. M., Chakrabortty, R. K., & Ryan, M. J. (2021). A reinforcement learning based multi-method approach for stochastic resource constrained project scheduling problems. Expert Systems with Applications, 169, 114479. https://doi.org/10.1016/j.eswa.2020.114479

[12] Zhang, Y., Zhang, Z., Lu, Y., Zhu, H., & Tang, D. (2024). Dynamic decision-making for knowledge-enabled distributed resource configuration in cloud manufacturing considering stochastic order arrival. Robotics and Computer-Integrated Manufacturing, 87, 87102712–. https://doi.org/10.1016/j.rcim.2023.102712

[13] Soltan, M. R., & Ashrafi, M. (2025). A new optimization model for multi-project scheduling considering dynamic resource allocation. Soft Computing, 29, 10891. https://doi.org/10.1007/s00500-025-10891-7

[14] Li, B., Wu, Y. (2025). Online reconfiguration for a hybrid line-seru production system: A two-stage optimization approach, Computers & Industrial Engineering, 209, 111481, https://doi.org/10.1016/j.cie.2025.111481.

[15] Barak, S., Javanmard, S., & Moghdani, R. (2024). Dual resource constrained flexible job shop scheduling with sequence-dependent setup time. Expert Systems, 41(10). https://doi.org/10.1111/exsy.13669

[16] Paraschos, P. D., & Koulouriotis, D. E. (2024). Learning-based production, maintenance, and quality optimization in smart manufacturing systems: A literature review and trends. Computers & Industrial Engineering, 198, 110656. https://doi.org/10.1016/j.cie.2024.110656

[17] Smendowski, M., & Nawrocki, P. (2024). Optimizing multi-time series forecasting for enhanced cloud resource utilization based on machine learning. Knowledge-Based Systems, 304, 304112489–304112489.https://doi.org/10.1016/j.knosys.2024.112489

[18] Morariu, C., Morariu, O., Răileanu, S., & Borangiu, T. (2020). Machine learning for predictive scheduling and resource allocation in large scale manufacturing systems. Computers in Industry, 120. https://doi.org/10.1016/j.compind.2020.103244

[19] Chen, X., Zhu, F., Chen, Z. et al. (2020), PP(99). Resource allocation for cloud-based software services using prediction-enabled feedback control with reinforcement learning. IEEE Transactions on Cloud Computing, 1–1. https://doi.org/10.1109/TCC.2020.2992537

[20] Wu, Y., Wu, Y. et al. Using extreme wind-speed probabilistic forecasts to optimize unit scheduling

decision. IEEE Transactions on Sustainable Energy. https://doi.org/10.1109/TSTE.2021.3132342