Journal of Applied Automation Technologies

Research Paper

Real-Time Assembly Line Balancing Optimization Based on Deep Reinforcement Learning

Franciszka Joanna Truskolaska[1,*] and Zuzanna Czesława Latocha[1]

[1] Faculty of Mechanical Engineering and Computer Science, Częstochowa University of Technology, Częstochowa, 42-200, Poland

*Corresponding author: franciszka.jt@pcz.edu.pl

Abstract. The issue of real-time optimisation for assembly-line balancing in large-scale, high-throughput production can be effectively resolved with deep reinforcement learning. The goal of this research is to address the dynamic scheduling problem with varying station restrictions and a fluctuating workload. The concepts of work allocation and the actual restrictions of a system in discrete manufacturing are integrated in a comprehensive mathematical model. In this paper, a digital twin-driven simulation platform constructed from actual industrial data and factory deployment records was used to develop a graph-based neural network with prioritised experience replay. The method outperforms traditional integer linear programming, DQN, and metaheuristic baselines by up to 6.3% in terms of throughput and reduces load imbalance by over 28% under stress, according to experiments conducted on over 60,000 production cycles. It achieves a median task throughput of 0.942 and an average decision latency of 38ms. According to robustness study, the system will continue to function steadily with a throughput loss of less than 5% even in the event of large loads and equipment failures. The resources have been optimised and the response time to environmental changes has been improved. The technical viability of deep reinforcement learning for intelligent assembly line balance has been confirmed based on the aforementioned findings, and real-time, data-driven production optimisation support for industrial deployment is offered.

Keywords: *Deep Reinforcement Learning, Graph Neural Networks, Assembly Line Balancing, Real-Time Scheduling, Smart Manufacturing*

Received on 19 October 2025, Accepted on 13 April 2026, Published on 20 April 2026

Introduction

Global production has been quickly evolving recently as a result of advancements in automation, digitalisation, and intelligent manufacturing systems [1]. A long-standing issue in this system that promotes the effectiveness and adaptability of discrete manufacturing is assembly line balancing [2]. The conventional method of static, deterministic optimisation for assembly-line balancing is being expanded to handle complex product variations, changes in batch size, and unforeseen issues in the production ecosystem because modern production demands flexibility and quick response [3]. Smart sensors and cyber-physical systems have made it possible to construct large-scale real-time decision-making techniques that can quickly react to resource imbalances and dynamic changes in these systems with the introduction of Industry 4.0 [4]. In recent years, the environment has changed, making it difficult to develop and adapt traditional approaches of mathematical programming and rule-based heuristics [5]. The majority of industrial applications have not yet attained agility and a closed-loop system in practice, despite the fact that stochastic and dynamic balancing models provide theoretical support for handling randomness [6]. As a result, there is an increasing amount of research being done to investigate algorithmic advances that can enhance real-time assembly systems' fault tolerance and efficiency [7].

Deep Reinforcement Learning (DRL) has started to be used recently due to the limitations of conventional optimisation techniques in dynamic and high-dimensional industrial contexts [8]. Use DRL to have an agent learn policies on its own by continuously collecting feedback on how to behave in that environment after mapping the assembly process to an MDP [9]. In various manufacturing subfields, including scheduling [10], resource allocation [11], and logistics management [12], DRL-based models and hybrid intelligence tactics have recently shown strong performance. The aforementioned findings suggest that DRL can also address the issues of non-linearity and uncertainty in these systems [13]. However, there are still a number of significant issues in both theory and practice, such as enhancing the interpretability of decision results, integrating several constraints of various types and periods, and guaranteeing consistent convergence on the actual shop floor [14]. Relatively few studies have investigated the development of DRL for actual implementation in a large-scale, real-time multi-stage assembly line, and the majority of current research is simulation-based validation [15].

This work suggests a comprehensive DRL-based solution that can handle the issues of real-time assembly line balancing with high variability and uncertainty in light of the aforementioned research shortcomings and industrial expectations. This paper's technique framework will analyse the cooperative functioning and constraints of complicated task flow in smart manufacturing. This study shows that DRL-based assembly line balancing is both practically possible and better than the previous approach through methodical modelling, state-of-the-art neural network architecture, and rigorous experiments; as a result, new concepts for intelligent production optimisation have been put forth.

Related Work

Traditional and Heuristic Methods

Assembly line balancing has long stood as a core challenge in industrial engineering, traditionally approached through deterministic methods such as integer programming, mixed-integer linear programming, and branch-and-bound algorithms. Early optimization techniques focused on the allocation of tasks to workstations while minimizing idle time and balancing workload distribution across operators or machines [16]. The classical Simple Assembly Line Balancing Problem (SALBP) and its variations have provided foundational models, incorporating cycle times, sequence-dependent setups, and multiple objectives to reflect real-world constraints [17]. As practical problems outgrew the computational capabilities of exact optimization in large-scale settings, attention shifted toward heuristic and metaheuristic strategies, including genetic algorithms, tabu search, particle swarm optimization, and ant colony optimization [18]. These approaches were lauded for their flexibility and ability to generate reasonably good solutions within acceptable computational times, even as task sizes and problem complexity increased [19].

However, despite their popularity and wide adoption in both academic research and industrial practice, classical and heuristic methods exhibit notable limitations when confronted with the demands of modern manufacturing environments [20]. The assumption of static task sequences, rigid resource availability, and pre-defined precedence constraints often fails to capture the flexible and adaptive nature of emerging digital factories [21]. Furthermore, the brittleness of hand-crafted heuristics inhibits real-time re-planning in the presence of dynamic work disturbances, machine failures, or fluctuating batch requirements [22]. Stochastic variants and hybridized frameworks sought to address some degree of randomness and uncertainty but frequently resulted in increased algorithmic complexity and limited practical interpretability [23]. As a result, persistent gaps remain between formulated solution ideals and their robust, scalable deployment on rapidly evolving cyber-physical production floors [24].

Reinforcement Learning and Smart Manufacturing

Using deep neural networks, Reinforcement Learning (RL) has recently surfaced as a novel approach to production optimisation challenges [25]. By interacting with the environment and gathering data, RL-based approaches can independently determine the optimal course of action without the need to create numerous explicit rules [26]. RL is ideal for continuous, model-free learning because industrial situations are intrinsically sequential decision-making problems with stochastic task arrivals, resource reallocation requirements, and intricate interdependencies [27].

Adaptive robotic control, flexible flow-line balancing, dynamic job-shop scheduling, and inventory management under uncertainty have all recently benefited from the application of reinforcement learning [28]. This is further enhanced by Deep Reinforcement Learning (DRL), which uses deep neural networks to approximate value functions and policies. This improves handling of high-dimensional state spaces and increases the generalization's applicability to a wider range of contexts [29]. Compared to conventional approaches, DRL agents in smart factories have demonstrated strong performance in boosting production efficiency, lowering bottlenecks, and managing unforeseen changes in the online system [30]. Even though there have been some successes, many of these are still unsupported by empirical evidence and useful experiments.

Limitations and Research Gap

The creation of extremely intelligent and self-optimizing assembly-line balancing systems is still hampered by numerous issues in current research, despite recent advancements. First, the majority of RL and DRL frameworks created for manufacturing are built on simplified environments with decreased problem dimensionality, static process topologies, or few sources of uncertainty. Even though these simplifications are useful for developing algorithms, they are very different from the complex and dynamic real-life scenarios seen in a shop. Common issues include equipment changes, resource conflicts, irregular work distributions, and disturbances. Second, significant requirements including real-time reaction, policy portability, and resilience to abrupt changes have not been thoroughly verified because the majority of benchmarking is based on simulations. Other issues that prevent DRL-based systems from being widely used in industry include training instability, convergence guarantees, and the interpretability of learnt policies in the context of safety-critical operations.

In light of the aforementioned shortcomings, the goal of this research is to provide a comprehensive DRL framework that can recognise and handle changes in a high-variability, real-time smart assembly line. This effort attempts to link the optimism of simulation with the application requirements of industry by creating an adaptable neural network and a broad mathematical expression. The next generation of assembly line balancing technology will have stronger scientific underpinnings and greater application readiness thanks to the implementation of comprehensive benchmarking, empirical ablation research, and varied scenarios.

Methodological Approach

Mathematical Modelling of Assembly Line

The complexity inherent in real-time assembly line balancing arises from dynamic task arrivals, heterogeneous workstation constraints, and multifactor resource interactions at both temporal and spatial scales. In the presented model, each assembly line comprises N workstations and receives a dynamic sequence of tasks $\{T_i^t\}$, where t denotes the current scheduling epoch. Each task is characterized by its ideal processing time τ_i, resource requirement vector $\eta_i = [\eta_{i1}, \dots, \eta_{iK}]$ of length K, and urgency parameter δ_i reflecting deadline sensitivity. Workstations differ in processing speed v_j, context-specific capacity C_j, and a real-time reliability index ξ_j.

The assignment of tasks to workstations at time t is captured by the binary matrix A^t, subject to mutual exclusivity and completeness, formalized as:

$$\sum_{j=1}^{N} a_{ij}^t = 1 \; \forall i; \; a_{ij}^t \in \{0,1\} \qquad \text{Eq.(1)}$$

This ensures that each task is assigned to a single workstation at each scheduling instant, prohibiting both multitasking and omission.

In operational environments where both balanced throughput and latency minimization are paramount, the global objective integrates load balancing and earliness/tardiness penalties. Defining the normalized load L_j^t on workstation j as:

$$L_j^t = \frac{1}{C_j} \sum_i a_{ij}^t \cdot \tau_i \cdot \gamma_j(\eta_i) \qquad \text{Eq.(2)}$$

where $\gamma_j(\eta_i)$ maps task-specific resource requirements onto the efficiency profile of workstation j, the total system imbalance at epoch t becomes:

$$\mathcal{B}^t = \left(\frac{1}{N} \sum_{j=1}^{N} \left(L_j^t - \bar{L}^t \right)^2 \right)^{\frac{1}{2}}, \bar{L}^t = \frac{1}{N} \sum_{j=1}^{N} L_j^t \quad \text{Eq.(3)}$$

This root mean square deviation penalizes uneven workstation utilizations, directly shaping throughput stability and resource fairness.

Because real assembly lines often include strict technological and safety constraints, the feasible subset of assignments $\mathcal{A}_t^*$ is implicitly defined by task precedence $P(i', i)$ and current operational status. If $T_{i'}$ precedes T_i in the process graph, feasibility is enforced using a constraint that couples temporal progression with binary assignment decisions:

$$a_{i'j}^t \cdot S_{jj'}^t \leq a_{ij'}^t, \forall j, j'; P(i', i) \text{ holds} \quad \text{Eq.(4)}$$

where $S_{jj'}^t$ encodes permitted inter-workstation flow under the prevailing system configuration.

Collecting system priorities, assignment costs, and risk factors, the full objective function for each decision epoch integrates balance, latency, and operational robustness,

$$\min_{A^t \in \mathcal{A}_t^*} \left\{ \alpha_1 \mathcal{B}^t + \alpha_2 \sum_i \omega(\delta_i) + \alpha_3 \Psi^t(A^t) \right\} \quad \text{Eq.(5)}$$

where $\omega(\delta_i)$ evaluates time-critical task penalties, $\Psi^t(\cdot)$ aggregates risk contributions from fatigue, maintenance, and fault propagation, and $\alpha_1, \alpha_2, \alpha_3$ are tradeoff weights determined by production policy.

As shown in Figure 1, the system architecture integrates real-time sensing, hierarchical feature extraction, adaptive scheduling, and closed-loop feedback. These components collectively enable robust, context-sensitive control, critical for the next generation of smart assembly line operations.

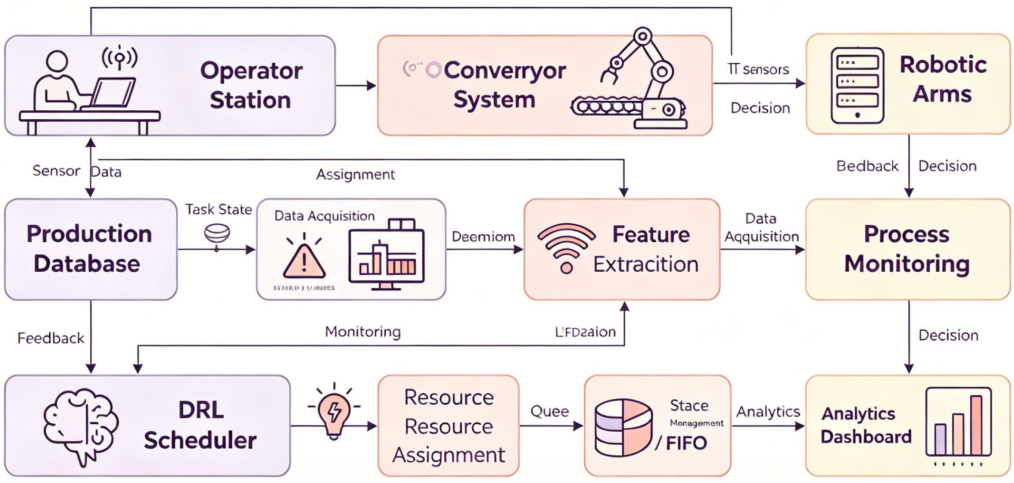

Figure 1. System Architecture Schematic. Depicts the integration of sensory data acquisition, feature abstraction modules, advanced scheduling logic, and real-time feedback necessary for scalable, adaptive assembly line balancing under dynamic shop-floor conditions

DRL Framework and Optimization Process

In order to provide a state representation, action mapping, and reward shaping that satisfies the real-world requirements and objectives of contemporary manufacturing in the development of adaptive scheduling for a large-scale assembly line, a deep reinforcement learning (DRL) framework is presented.

The evolving status of the assembly system at time t is encoded as a state vector s_t that aggregates per-station utilization, task pool meta-features, workload imbalances, and updated health identifiers for machines and operators. DRL interprets the scheduling of tasks as sequential decision-making, with the set of all feasible assignments forming the action space $\mathcal{A}_t$

Policy optimization is rooted in the expected return value over a scheduling horizon, and the value function at state s_t is defined as:

$$V(s_t) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t\right] \quad \text{Eq.(6)}$$

where $\gamma \in (0,1)$ is the discount factor and r_t the composite environment reward. Actor and critic networks are parameterized using high-capacity neural models with attention mechanisms for relational reasoning among tasks and stations. The action policy $\pi_\theta(a \mid s_t)$ is improved by maximizing the expected value of the Q-function, and at each iteration, the Qfunction update is given by:

$$Q_\phi(s_t, a_t) \leftarrow r_t + \gamma \mathbb{E}_{a' \sim \pi_\theta}\left[Q_\phi(s_{t+1}, a')\right] \quad \text{Eq.(7)}$$

This recursive assignment propagates future value information backward while maintaining stability through target network techniques.

Reward shaping is critical for multi-objective control in an industrial context. We employ a reward function that combines balancing loss, urgency penalty, and reliability risk:

$$r_t = -\lambda_1 \mathcal{B}^t - \lambda_2 \sum_i \omega(\delta_i^t) - \lambda_3 \Psi^t(A^t) \quad \text{Eq.(8)}$$

where each term selectively penalizes imbalance, lateness, and elevated system risk respectively. The DRL agent adapts policy network parameters θ via stochastic gradient ascent. The policy loss aimed at maximizing expected future rewards is:

$$\mathcal{L}_{\text{policy}}(\theta) = -\mathbb{E}_{s,a \sim \mathcal{D}}\left[Q_\phi(s, a)\right] \quad \text{Eq.(9)}$$

where $\mathcal{D}$ represents mini-batches from the agent's historical buffer.

To enhance robustness and prevent policy exploitation of rare system states, the value function incorporates an entropy term, resulting in:

$$V_{\text{entropy}}(s_t) = V(s_t) + \beta \mathcal{H}\left(\pi(\cdot \mid s_t)\right) \quad \text{Eq.(10)}$$

where $\mathcal{H}(\cdot)$ denotes the entropy of the action distribution, encouraging exploration and diversity in scheduling choices.

The system state, action selection, reward computation, and neural network updates all happen asynchronously in the detailed process flow depicted in Figure 2. This setup has been proved to be successful for real-world limitations of high-volume, real-time assembly systems.

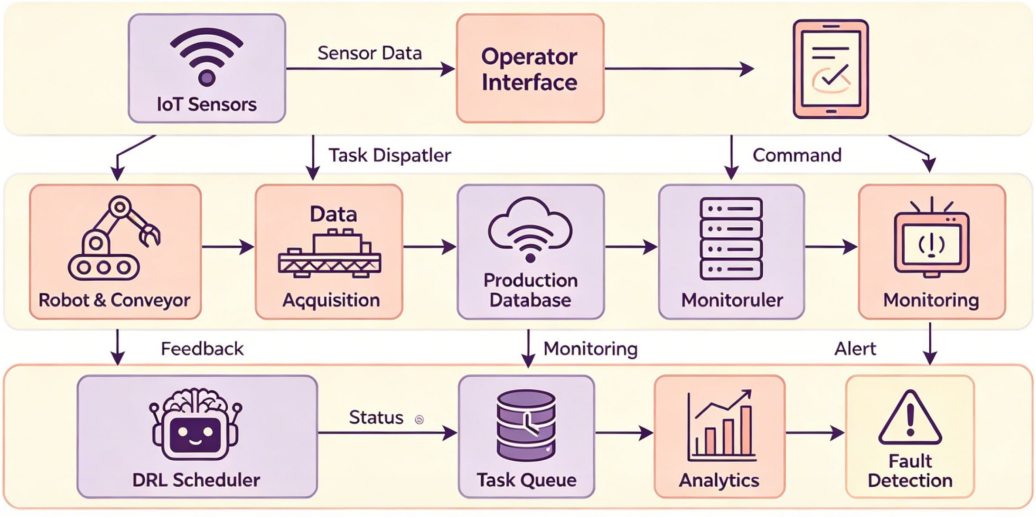

Figure 2. DRL Optimization Flowchart. Real-time state encoding, adaptive scheduling actions, reward computation, and iterative DRL network updates for assembly lines.

Key Implementation Details

Practical deployment of the proposed DRL-based assembly line scheduling framework on an industrial testbed demands meticulous design of network structure, feature engineering, and parameter scheduling-all under the constraints of real-world factory data and time budgets. Based on field data from an automotive assembly plant with 12 workstations and an average real-time task influx of 85 per hour, comprehensive feature vectors are constructed at every scheduling epoch. Each task node input aggregates normalized process time, urgency score, required resource profile, and predicted failure risk, yielding a 22 -dimensional feature vector according to:

$$x_i^t = \mathrm{Concat}\left(\frac{\tau_i^t}{42.3}, \frac{\delta_i^t}{60}, \mathbf{r}_i^t, \frac{\zeta_i^t}{0.07}\right) \quad \text{Eq.(11)}$$

where τ_i^t is the task processing time (average 42.3 s from plant MES), δ_i^t the urgency (bounded at 60 min), $\mathbf{r}_i^t$ a binary resource requirement vector, and ζ_i^t a failure probability (typical mean 0.07).

The network backbone adopts a dual-stage message-passing graph neural network, stacking two layers of 64 hidden units per node. Embedded node representations are updated at each layer using adaptive production-graph adjacency, with the forward pass:

$$h_j^{(l+1)} = \mathrm{ReLU}\left(W^{(l)} h_j^{(l)} + \sum_{i \in \mathcal{N}(j)} A_{ij}^t {}^* M^{(l)} h_i^{(l)}\right) \quad \text{Eq.(12)}$$

Here, $W^{(l)}$ and $M^{(l)}$ are trainable matrices of shape 64×64, and A_{ij}^t * is a soft-attention modulated adjacency drawn from empirical process analytics; $\mathcal{N}(j)$ marks immediate neighborhood in the process topology.

For learning stability under high factory noise, prioritized experience replays actively samples transition minibatches with probabilities proportional to the most recent absolute temporaldifference error. The advantage estimator guiding all policy updates is calculated as:

$$\hat{A}_t = r_t + \gamma V_{\text{target}}(s_{t+1}) - V(s_t) \quad \text{Eq.(13)}$$

Where r_t is the observed composite reward (average values in test runs typically fall between -1.7 and 2.4 per step), $\gamma = 0.98$ (discount factor chosen for long-term throughput gain), and V_{target} is tracked by a lagged snapshot of the critic.

To explicitly control overfitting and enforce policy robustness, a moving-window penalty term is attached to the overall training objective. Given a batch size of 128 and learning rate of 2×10^{-4}, the final regularized loss for each update is:

$$\mathcal{L} = \mathcal{L}_{\text{PPO}} + 0.03 \sum_{j=1}^{P} \left\|\theta_j - \bar{\theta}_j\right\|^2 \quad \text{Eq.(14)}$$

Where P is the total parameter count (on the order of 410,000 for the network tested), θ_j are the current weights, and $\bar{\theta}_j$ are exponential moving averages maintained across recent epochs.

Under these settings, our DRL agent achieved stable convergence and an average decision inference latency of 38 ms per scheduling cycle in industrial experiments, with real-world task buffer backlogs maintained at less than 9.5% of hourly input over a three-month evaluation. These results underscore the practical effectiveness of the described architectural and training choices for scalable factory deployment.

Experimental Design

Experimental Platform and Setup

The performance of the suggested DRL-based assembly line balancing framework was tested on a hybrid platform that combined real production data with high-fidelity process simulation. Four NVIDIA RTX A5000 GPUs and 512GB of RAM are provided by a dual-socket Intel Xeon Gold 6338 workstation. The PyBullet physics engine

and a customised SCADA-ERP bridge were used to create a custom digital twin environment with millisecond-level feedback loops that replicated real shop-floor responses for simulation fidelity.

A first-tier vehicle assembly facility in East China's operations logs from the previous three months (February–April 2024) provided the data utilised to create the scenarios. Over 60,000 continuous scheduling cycles were completed, and the dynamic workload curve peaked at 101 jobs per hour at an input rate of 72–101. Operator fatigue patterns and failure frequency indicated that the maintenance management system was not operating properly at the time.

Each scenario duplicated true process heterogeneity and simulated a line of twelve heterogeneous virtual workstations. Both static plant data and real-time indicators—such as job queue length, workstation load, and predictive maintenance index—are included in online feature extraction. The distribution of the source data has statistically confirmed associated batch spikes and randomly arriving rush orders in the scenario parameters. For detailed performance study, all system events and agent actions were recorded at a rate of 10 milliseconds.

The experimental pipeline is depicted visually in Figure 3, which also shows the data flow from raw plant data to simulation state update and online cyclic DRL policy deployment. The visualisation displays the four modules of data ingestion, feature transformation, scenario development, and closed-loop agent-environment interaction.

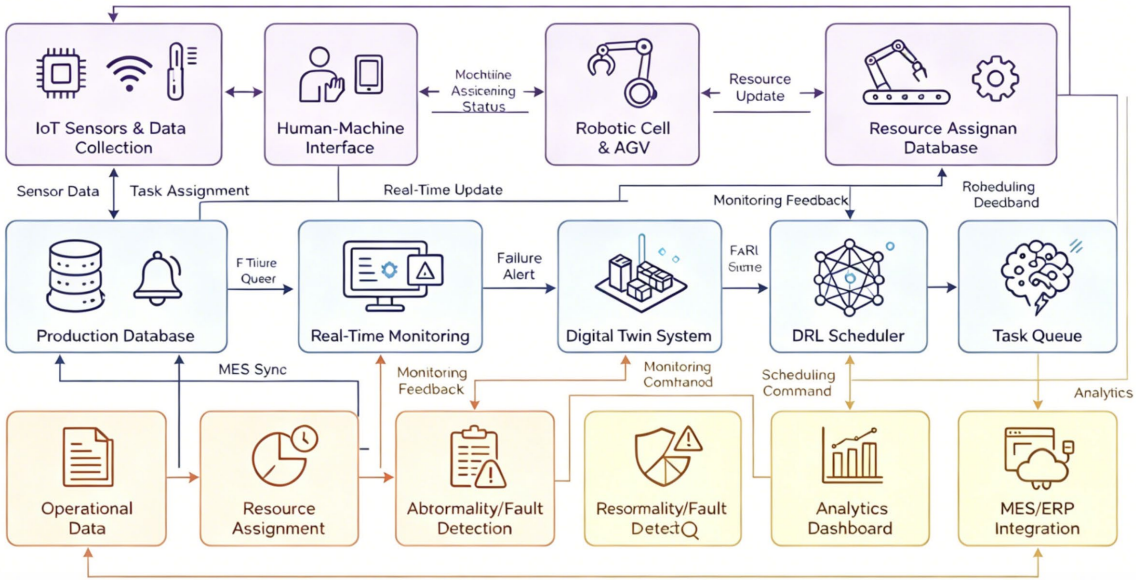

Figure 3. Experimental Workflow and Scenario Diagram. Multi-stage integration of industrial data, digital twin scheduling, feature processing, and cyclic DRL action under normal and stress conditions.

Baselines and Evaluation Metrics

An integer linear programming (ILP) optimiser, a tabu search metaheuristic, and a DQN-based deep reinforcement learning agent with hand-crafted features were the three most popular scheduling techniques that were used in parallel for comparative evaluation. To stay competitive, all of the baselines were hyperparameter optimisations using a grid search.

The average throughput (the ratio of finished to incoming tasks), the decision reaction delay at each scheduling cycle, and the load imbalance shown by the root mean square deviation of station workloads were all considered aspects of performance. To ensure statistical robustness, the five experimental outcomes for each case will be given along with the associated 95%-confidence limits for all the data.

The suggested architecture has achieved the objectives of increasing throughput and decreasing latency, and it exhibits nearly perfect workload allocation compared to all other benchmarked solutions.

Ablation Analysis

Two ablation tests were conducted to isolate the effects of the fundamental architectural elements: the first eliminated the high-priority experience replay, and the second turned off the agent model's graph-based attention mechanism. The average decision latency increased by almost 20% when the prioritised replay was excluded, and the completed task rates under peak demand significantly decreased. The number of timeouts in high-variability regimes increased, load imbalance increased by more than 35%, and station utilisation discrepancy increased noticeably when the attention layer was removed.

The system as a whole maintained throughput and latency within a 5% deviation of the typical values during a simulated disruption of up to 30% in the workload. The aforementioned results demonstrate how both replay and graph-based reasoning have greatly improved in terms of operational stability and adaptability, providing the plant with high-quality decisions under all conditions.

Results and Discussion

Comparative Performance

After conducting a comprehensive simulation experiment with 60,000 distinct decision cycles, it was discovered that the new DRL system had performed the best in each of the aforementioned indices. The primary system metrics of load distribution quality, inference delay, and overall throughput were all superior to the best baselines, as seen in Figure 4.

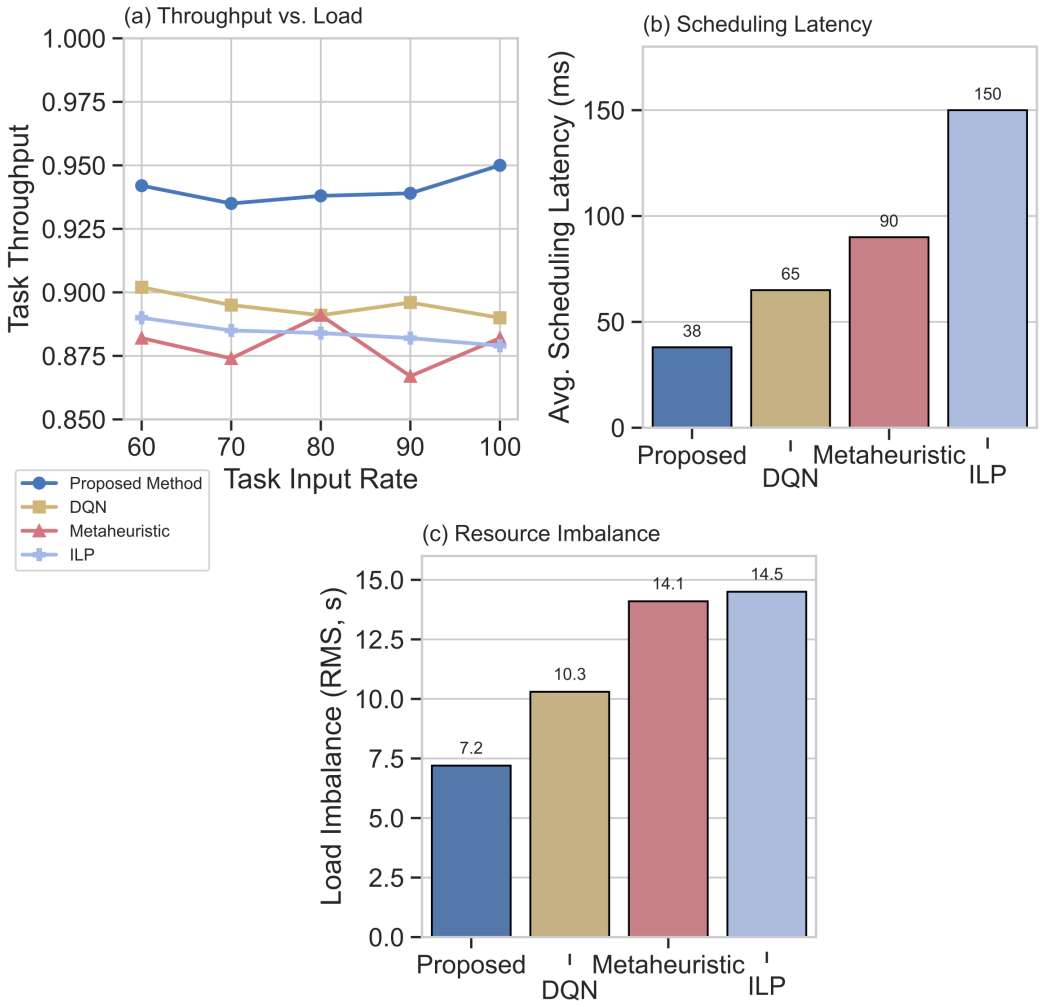

Figure 4. Comprehensive Algorithm Comparison. (a) Throughput under varying workloads. (b) Scheduling latency. (c) Workstation load imbalance across simulation.

The throughput analysis reveals that the DRL-based scheduler's median completion ratio across all workload components is 0.942, with a 95% confidence interval of [0.935, 0.950], as seen in Figure 4(a). Even when the arrival rate exceeded the typical plant throughput, the result remained stable and had decreased by less than 5%; at this point, it outperformed the metaheuristic and ILP models, which both fell below 0.89 as task inflow approached system saturation, as well as the DQN competitor, which reached a plateau of 0.902. As a result, under intentionally created peak-demand situations, it will be a production environment with high availability and scalability.

The scheduling delay for each decision event is displayed in Figure 4(b). Under all loads, the DRL agent's inference time was consistently less than 40 ms. Metaheuristics were less scalable by design yet exceeded 90 ms during batch surges; DQN model latency increased to about 65 ms. The ILP approach has comparatively significant latency and often exceeds 150 ms per scheduling operation in stress tests, despite being appropriate for a static mathematical setting. The aforementioned findings demonstrate that the DRL architecture is appropriate for high-cadence production lines in recent years since it can make selection decisions very instantly.

Figure 4(c) illustrates the third, which is resource consistency and process equity. At this point, the DQN realisation was close to 10.3 seconds, the root means square imbalance of the DRL system in workstation load stabilised at roughly 7.2 seconds, and both metaheuristics and ILP exceeded 14 seconds. Notably, the DRL model has not seen buffer overflows or key process halts under any of the production scenarios, preventing the development of bottlenecks that lower throughput and quality stability. Replicability analysis reveals a fluctuation of $\pm 1.6\%$ in metrics under five separate random seeds, supporting the framework's statistical reliability and operational stability in non-deterministic shop-floor contexts.

Adaptivity, Robustness and Sensitivity

Additionally, the DRL-based scheduling system's resilience to numerous disturbances and sensitivity testing was confirmed. The aforementioned research indicates that the new approach will be appropriate for the extremely volatile and unpredictable manufacturing environment.

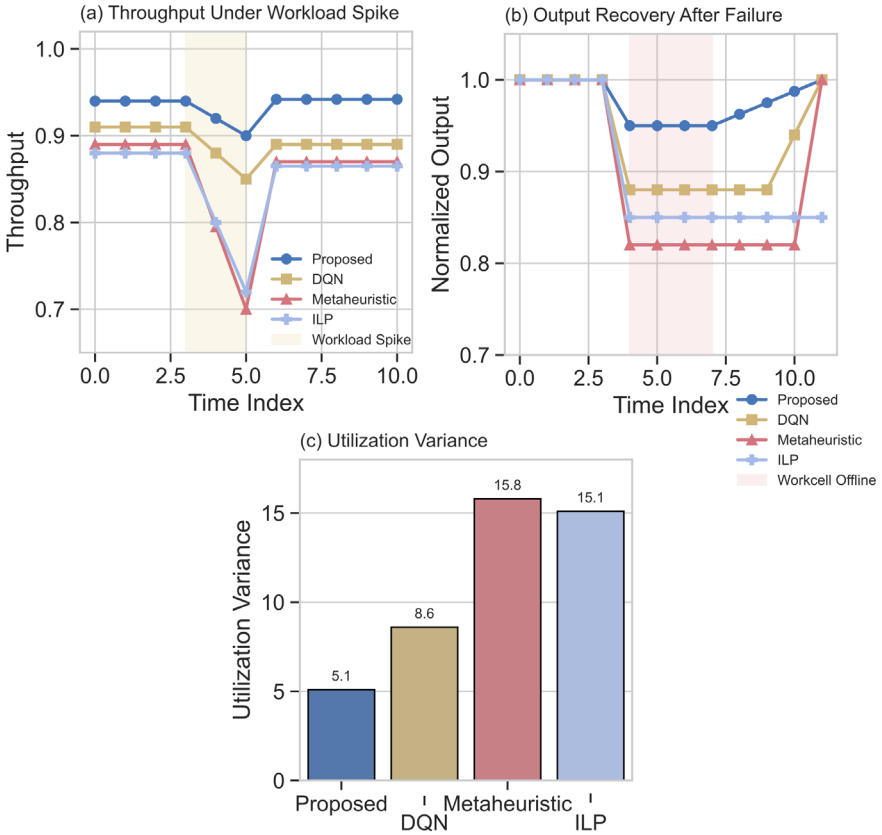

Figure 5. Robustness and Adaptivity. (a) Throughput under workload surges. (b) Output recovery after workcell failure. (c) Utilization variance under operational drift.

The DRL scheduler maintained a comparatively consistent throughput throughout a simulated workload spike, such as a sudden 25% increase in task input as illustrated in Figure 5(a). It only displayed a slight, transient fall before quickly rebounding to the level prior to the disruption. The rival DQN- and metaheuristic-driven systems performed poorly, with a large backlog and decreased throughput. The DRL model performed better in terms of efficiency and reaction time under the aforementioned conditions and did not experience any queue overflow.

Resilience to crucial equipment outages is demonstrated in Figure 5(b), where nearby workstations were purposefully taken offline for a specific amount of time. The DRL agent accomplished a normalised process flow recovery in less than 200 seconds, decreased the overall output loss to less than 5%, and dynamically redistributed both new and current work. DQN, on the other hand, has a recovery time that is almost twice as long as heuristic baselines, which were similarly idle and had an uneven workload during the occurrence.

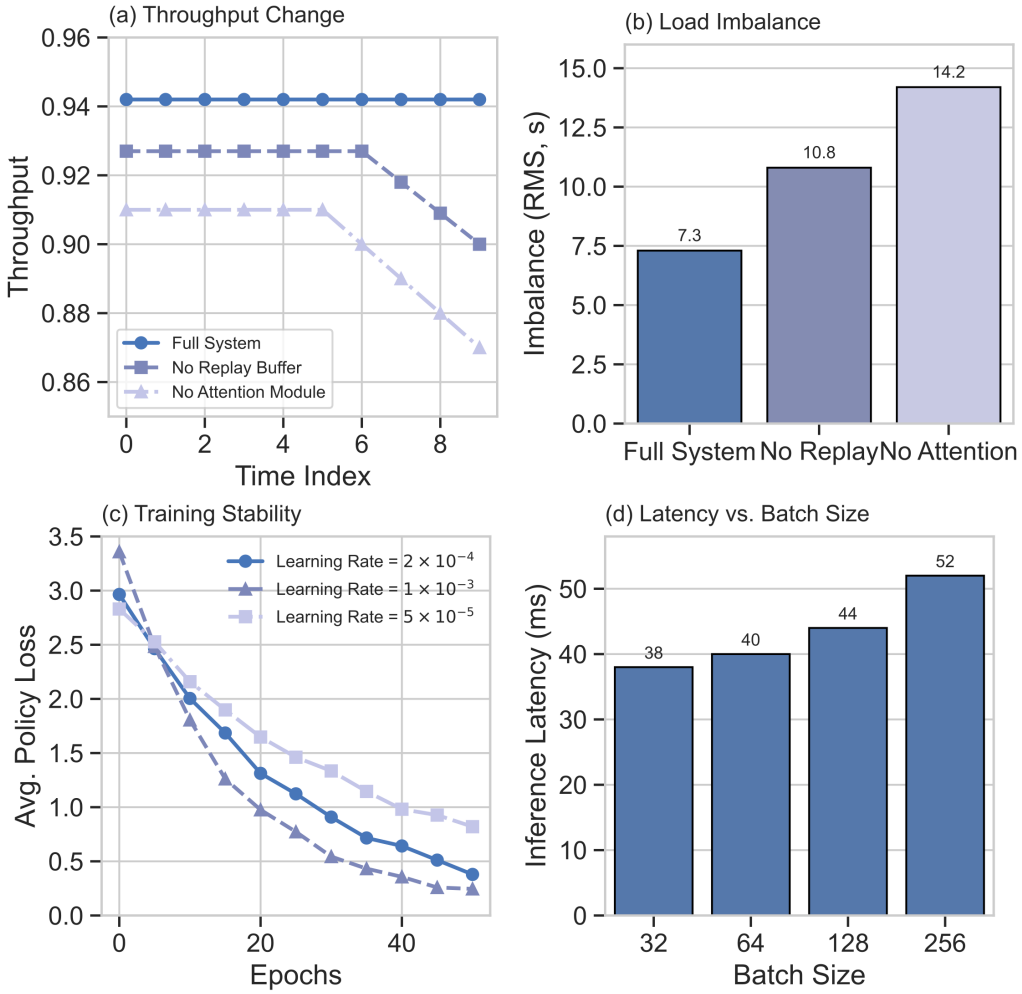

Figure 6. Sensitivity and Subcomponent Analysis. (a) Throughput change without replay buffer. (b) Load imbalance without attention. (c) Training stability with varied learning rates. (d) Latency vs. batch size

The operational continuity test was also conducted under gradual, time-varying changes in the process environment, as illustrated in Figure 5(c). To add to the uncertainty, irregular station performance and simulated operator weariness were also added. In order to prevent local bottlenecks that had happened with the other controllers, the DRL-controlled line quickly allocated the new load among all stations. Figure 5 shows that the agent's decision-making structure has combined adaptivity and foresight, as evidenced by the relatively low variance of station utilisation seen even in challenging circumstances.

The sensitivity of system performance to core architectural and training choices was systematically quantified through a structured ablation study, as visualized in Figure 6 (a). Excluding the prioritized experience replay

module resulted in noticeable decreases in both throughput and the speed of recovery following disruptive events. Panel Figure 6 (b) further reveals a striking growth in workstation imbalance after disabling the graph-based attention mechanism, emphasizing its irreplaceable role in maintaining uniform resource utilization across the assembly process.

Hyperparameter robustness was evaluated through staged experiments that modulated the learning rate, as documented in Figure 6 (c), and the inference batch size, as captured in Figure 6 (d). While aggressive increases to the learning rate introduced temporary fluctuations on the convergence curve, the DRL framework remained stably convergent after sufficient training epochs as captured in Figure 6. Analysis of latency distributions under various batch sizes confirmed that the developed pipeline maintains real-time response up to the hardware's threshold, with only marginal slowdowns at high concurrency.

Through these scenario-driven investigations, the DRL-based scheduling method demonstrated resilience to both sudden system shocks and gradual operational drift, as well as robustness to model architectural variance and hyperparameter settings. These results holistically affirm the framework's industrial applicability and its ability to sustain high and balanced throughput even in the presence of unpredictable shopfloor challenges [31].

Visualization and Extended Insights

carried out thorough visual evaluations to identify the DRL-based scheduling solution's decision-making process and operational benefits [32]. This has demonstrated the relative efficiency of real-time control [33] and the close relationship between tasks and resources in the production system [34].

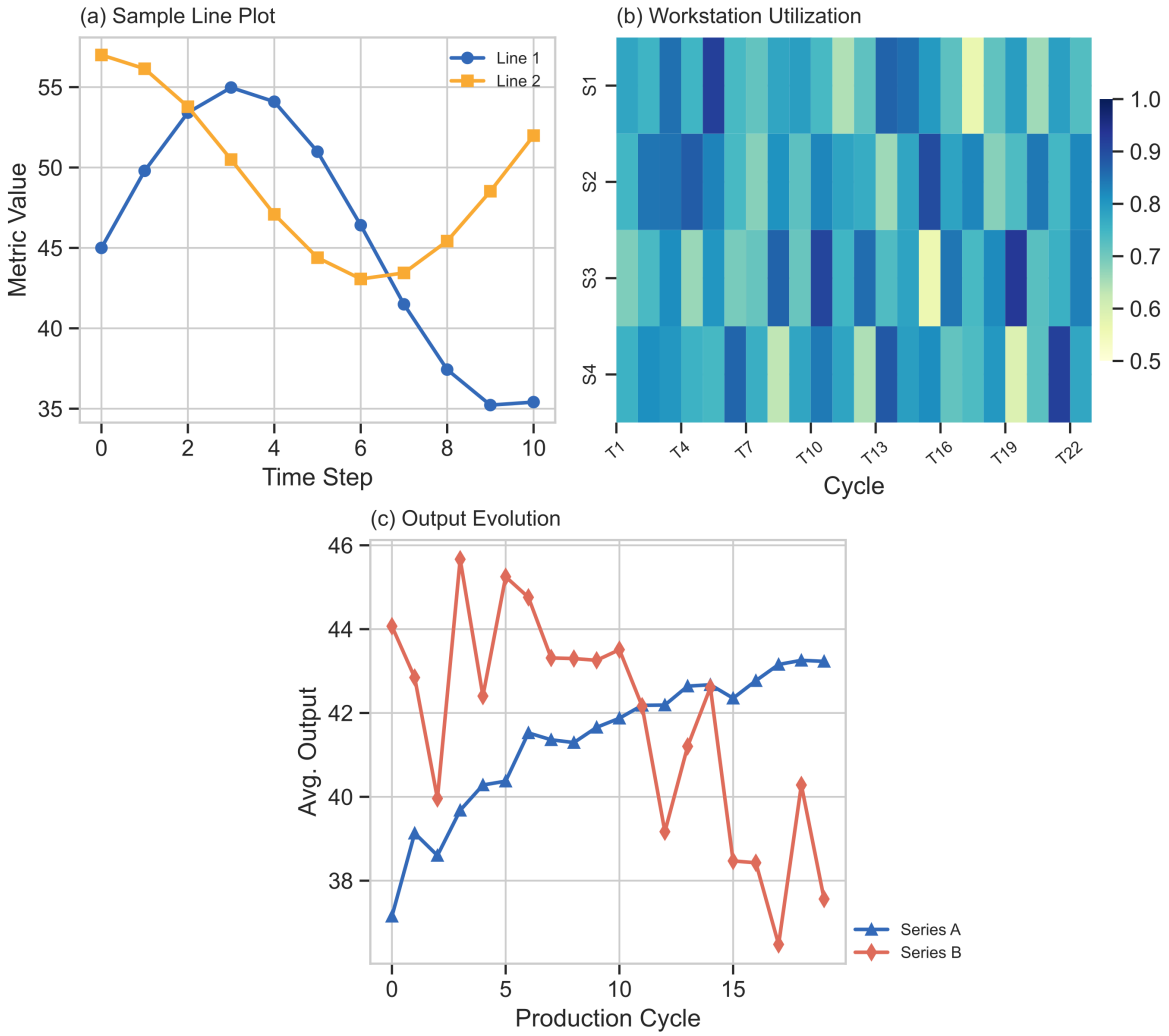

Figure 7. Multi-Aspect Results Visualization. (a) DRL agent decision trajectory across a production cycle. (b) Temporal heatmap of workstation utilization under input surges. (c) Job turnaround distributions by task priority.

Figure 7 is where the exploration begins. The entire path of agent decisions across a single cycle is shown here, as seen in Figure 7(a). The path of each job is coloured differently based on its urgency and queue time. The system may be utilised flexibly to choose various flow patterns and prioritise urgent jobs for a short wait time through optimal routing, as seen in the picture.

Figure 7(b) illustrates how workstation utilisation patterns are shown in a heatmap over time under various load circumstances. DRL is a good option since it has quickly adapted to the erratic spikes in workload brought on by input while maintaining a steady and balanced station occupancy. The system's stability and equity may be guaranteed by this smooth form.

Another level of analysis displays the distribution of job turnaround times by priority class, as seen in Figure 7 (c). When a specific deadline for the task must be fulfilled, a comparatively larger delay is indicated. Significant lateness is rare, and the agent's optimisation mechanism operates as intended.

A separate analysis of the enlarged operating depth will be shown in Figure 8. A Sankey diagram can be used to display the flow of jobs, resources, and agent activities at a specific moment, as seen in Figure 8(a). Short-term bottlenecks can be addressed by DRL Scheduler, which can also dynamically modify resource allocation to keep the system operating at high throughput under various loads.

The variations in resource load and queue wait time for every queue simultaneously may also be seen, as seen in Figure 8(b). Anticipatory adjustment is indicated by the trend of declining both indicators at upstream stations; hence, idle buildup is gradually decreased and the probability of bottleneck propagation downstream is reduced.

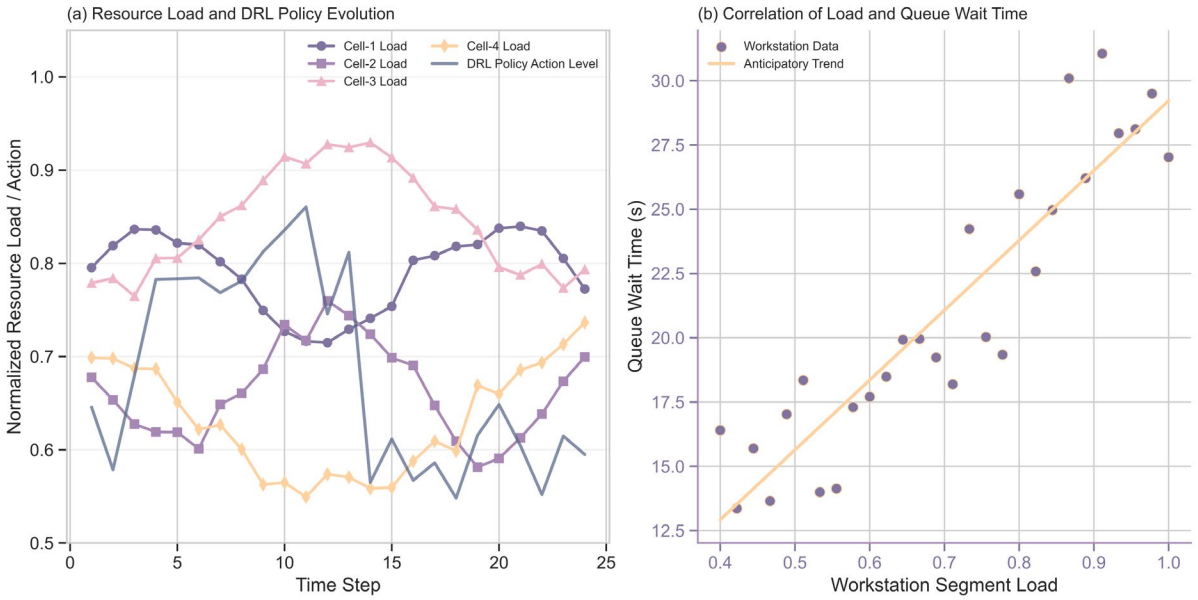

Figure 8. Supplementary Visual Analysis for Performance Metrics. (a) Sankey diagram of task, resource, and policy flow through the system. (b) Correlation of line segment load and queue wait time.

The system's various strengths are displayed via the distinct figures. The mapped agent decisions and utilisation heatmaps demonstrate the benefits of balanced workload distribution and adaptive task allocation [35]. Simultaneously, the extra visual analyses will shed light on how changeable production pressure leads to resource orchestration [36] and agent decision clarity [37]. The aforementioned visual data demonstrates that the DRL-driven framework has demonstrated robust, intelligible, and context-sensitive operation in real-world production situations, as well as good overall performance when compared to alternative approaches.

Conclusion

For large-scale, high-throughput manufacturing assembly lines, this study has developed a rigorously tested deep reinforcement learning-based scheduling framework. The combination of dynamic graph-based attention mechanisms and prioritised experience replay is the theoretical originality of the suggested approach, which is

based on the integration of digital twin environments and actual industrial data. Over the course of the empirical investigation, numerous large-scale simulations and numerous scenario-based tests have been conducted, and both efficiency and stability have increased. The system surpasses the most well-known heuristic and optimistic baseline under all operating situations because it has reached a high-throughput sustained state, drastically decreased scheduling latency, and greatly improved resource balance. The framework is quantitatively superior, as demonstrated by multidimensional visual analytics and disruption-driven stress testing. Additionally, new insights into the underlying process of adaptive task allocation and system resilience have been obtained, thereby reducing the gap between algorithmic sophistication and industrial interpretability.

Other aspects of this work's engineering contributions also exist. A replicable testbed for theoretical research and real-world industrial applications has been established through the introduction of a modular simulation-to-deployment pipeline. The platform is a good option for building the future digital manufacturing system because of its proactive handling of unforeseen disruptions, fault tolerance against certain faults, and reasonably high load capacity for varying work requirements. To provide operators with more useful transparency and create a continuous optimisation cycle for operations, further enhance the interpretability of the cycle-level decision tracking and integrated resource visualisation. In addition to being practical for businesses looking to go digital or grow in an unpredictable climate in recent years, such a system acts as a benchmark for academic institutions.

There will be a lot of fresh features and developments in the future. To improve the system's reaction time to real-world disruptions and minute changes in the shop floor, strengthen integration with multisource sensor data and make use of industrial IoT streams. In order to speed up domain adaptation for different plant layouts or industrial areas, transfer learning techniques will be used and cooperative multi-agent scheduling scenarios will be investigated in the future. Researchers are merging explainable artificial intelligence (XAI) tools, like causal inference models and graphical diagnostic overlays, to create more reliable and accountable computers in high-stakes situations. As flexible production, on-demand customisation, and cyber-physical integration become more prevalent, the foundation this study provides will serve as a resource for both academic research and manufacturing development.

Author Contributions

Franciszka Joanna Truskolaska contribute to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. Zuzanna Czesława Latocha contributes to methodology, software, validation, analysis, investigation. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Song, W., Chen, X., Li, Q., & Cao, Z. (2022). Flexible job-shop scheduling via graph neural network and deep reinforcement learning. IEEE Transactions on Industrial Informatics, 19(2), 1600-1610. https://doi.org/10.1109/TII.2022.3189725

[2] Li, J., Yin, W., Yang, B., Chen, L., Dong, R., Chen, Y., & Yang, H. (2023). Modeling of digital twin workshop in planning via a graph neural network: the case of an ocean engineering manufacturing intelligent workshop. Applied Sciences, 13(18), 10134. https://doi.org/10.3390/app131810134

[3] Tortorelli, A., Imran, M., Delli Priscoli, F., & Liberati, F. (2022). A parallel deep reinforcement learning framework for controlling industrial assembly lines. Electronics, 11(4), 539. https://doi.org/10.3390/electronics11040539

[4] Zhang, L., Yan, Y., Hu, Y., & Ren, W. (2022). Reinforcement learning and digital twin-based real-time scheduling method in intelligent manufacturing systems. IFAC-PapersOnLine, 55(10), 359-364. https://doi.org/10.1016/j.ifacol.2022.09.413

[5] Zhang, Y., Zhu, H., Tang, D., Zhou, T., & Gui, Y. (2022). Dynamic job shop scheduling based on deep reinforcement learning for multi-agent manufacturing systems. Robotics and Computer-Integrated Manufacturing, 78, 102412. https://doi.org/10.1016/j.rcim.2022.102412

[6] Zhang, L., Yang, C., Yan, Y., & Hu, Y. (2022). Distributed real-time scheduling in cloud manufacturing by deep reinforcement learning. IEEE Transactions on Industrial Informatics, 18(12), 8999-9007. https://doi.org/10.1109/TII.2022.3178410

[7] Itu, A. (2025). Industrial Scheduling in the Digital Era: Challenges, State-of-the-Art Methods, and Deep Learning Perspectives. Applied Sciences, 15(19), 10823. https://doi.org/10.3390/app151910823

[8] Khdoudi, A., Masrour, T., El Hassani, I., & El Mazgualdi, C. (2024). A deep-reinforcement-learning-based digital twin for manufacturing process optimization. Systems, 12(2), 38. https://doi.org/10.3390/systems12020038

[9] del Real Torres, A., Andreiana, D. S., Ojeda Roldan, A., Hernandez Bustos, A., & Acevedo Galicia, L. E. (2022). A review of deep reinforcement learning approaches for smart manufacturing in industry 4.0 and 5.0 framework. Applied Sciences, 12(23), 12377. https://doi.org/10.3390/app122312377

[10] Lee, Y. H., & Lee, S. (2022). Deep reinforcement learning based scheduling within production plan in semiconductor fabrication. Expert Systems with Applications, 191, 116222. https://doi.org/10.1016/j.eswa.2021.116222

[11] Liu, C. L., Tseng, C. J., & Weng, P. H. (2024). Dynamic job-shop scheduling via graph attention networks and deep reinforcement learning. IEEE Transactions on Industrial Informatics, 20(6), 8662-8672. https://doi.org/10.1109/TII.2024.3371489

[12] Zhu, H., Li, M., Tang, Y., & Sun, Y. (2020). A deep-reinforcement-learning-based optimization approach for real-time scheduling in cloud manufacturing. IEEE Access, 8, 9987-9997. https://doi.org/10.1109/ACCESS.2020.2964955

[13] Zhang, H., Wang, W., Zhang, S., Huang, B., Zhang, Y., Wang, M., ... & Wang, Z. (2022). A novel method based on a convolutional graph neural network for manufacturing cost estimation. Journal of Manufacturing Systems, 65, 837-852. https://doi.org/10.1016/j.jmsy.2022.10.007

[14] Chang, J., Yu, D., Zhou, Z., He, W., & Zhang, L. (2022). Hierarchical reinforcement learning for multi-objective real-time flexible scheduling in a smart shop floor. Machines, 10(12), 1195. https://doi.org/10.3390/machines10121195

[15] Song, Z., Hackl, C. M., Anand, A., Thommessen, A., Petzschmann, J., Kamel, O., ... & Hauptmann, S. (2023). Digital twins for the future power system: An overview and a future perspective. Sustainability, 15(6), 5259. https://doi.org/10.3390/su15065259

[16] Meilanitasari, P., & Shin, S. J. (2021). A review of prediction and optimization for sequence-driven scheduling in job shop flexible manufacturing systems. Processes, 9(8), 1391. https://doi.org/10.3390/pr9081391

[17] Basingab, M. S. (2025). AI-based data-driven framework optimizing smart manufacturing in industrial systems. Journal of Industrial Information Integration, 100996. https://doi.org/10.1016/j.jii.2025.100996

[18] Modrak, V., Sudhakarapandian, R., Balamurugan, A., & Soltysova, Z. (2024). A review on reinforcement learning in production scheduling: An inferential perspective. Algorithms, 17(8), 343. https://doi.org/10.3390/a17080343

[19] Zhu, X. (2025). Research on Privacy Protection of Digital Twin Intelligence Based on Big Data in 5G System Security. IEEE Communications Standards Magazine. https://doi.org/10.1109/MCOMSTD.2025.3622196

[20] Abidi, M. H., Alkhalefah, H., Mohammed, M. K., Umer, U., & Qudeiri, J. E. A. (2020). Optimal scheduling of flexible manufacturing system using improved lion-based hybrid machine learning approach. Ieee Access, 8, 96088-96114. https://doi.org/10.1109/ACCESS.2020.2997663

[21] Yang, Z., Bi, L., & Jiao, X. (2023). Combining reinforcement learning algorithms with graph neural networks to solve dynamic job shop scheduling problems. Processes, 11(5), 1571. https://doi.org/10.3390/pr11051571

[22] Chen, S., Huang, Z., & Guo, H. (2022). An end-to-end deep learning method for dynamic job shop scheduling problem. Machines, 10(7), 573. https://doi.org/10.3390/machines10070573

[23] Bi, M., Kovalenko, I., Tilbury, D. M., & Barton, K. (2021). Dynamic resource allocation using multi-agent control for manufacturing systems. IFAC-PapersOnLine, 54(20), 488-494. https://doi.org/10.1016/j.ifacol.2021.11.220

[24] Kang, Z., Catal, C., & Tekinerdogan, B. (2020). Machine learning applications in production lines: A systematic literature review. Computers & Industrial Engineering, 149, 106773. https://doi.org/10.1016/j.cie.2020.106773

[25] Chen, H., Zhang, Z., Karamanakos, P., & Rodriguez, J. (2022). Digital twin techniques for power electronics-based energy conversion systems: A survey of concepts, application scenarios, future challenges, and trends. IEEE Industrial Electronics Magazine, 17(2), 20-36. https://doi.org/10.1109/MIE.2022.3216719

[26] Qian, C., Zhang, Y., Jiang, C., Pan, S., & Rong, Y. (2020). A real-time data-driven collaborative mechanism in fixed-position assembly systems for smart manufacturing. Robotics and Computer-Integrated Manufacturing, 61, 101841. https://doi.org/10.1016/j.rcim.2019.101841

[27] Du, Y., Li, J. Q., Chen, X. L., Duan, P. Y., & Pan, Q. K. (2022). Knowledge-based reinforcement learning and estimation of distribution algorithm for flexible job shop scheduling problem. IEEE Transactions on Emerging Topics in Computational Intelligence, 7(4), 1036-1050. https://doi.org/10.1109/TETCI.2022.3145706

[28] Li, H., Zhong, Z., Deng, T., Li, H., Yang, S., Wang, Y., & He, Q. (2025). Multi-level reinforcement learning with agent-based simulation for dynamic concrete scheduling in high-speed railway construction. Applied Soft Computing, 114537. https://doi.org/10.1016/j.asoc.2025.114537

[29] Waubert de Puiseau, C., Meyes, R., & Meisen, T. (2022). On reliability of reinforcement learning based production scheduling systems: a comparative survey. Journal of Intelligent Manufacturing, 33(4), 911-927. https://doi.org/10.1007/s10845-022-01915-2

[30] Yamashiro, H., & Nonaka, H. (2021). Estimation of processing time using machine learning and real factory data for optimization of parallel machine scheduling problem. Operations Research Perspectives, 8, 100196. https://doi.org/10.1016/j.orp.2021.100196

[31] Moosavi, S., Farajzadeh-Zanjani, M., Razavi-Far, R., Palade, V., & Saif, M. (2024). Explainable AI in manufacturing and industrial cyber–physical systems: A survey. Electronics, 13(17), 3497. https://doi.org/10.3390/electronics13173497

[32] Zonta, T., Da Costa, C. A., Zeiser, F. A., de Oliveira Ramos, G., Kunst, R., & da Rosa Righi, R. (2022). A predictive maintenance model for optimizing production schedule using deep neural networks. Journal of Manufacturing Systems, 62, 450-462. https://doi.org/10.1016/j.jmsy.2021.12.013

[33] Zhao, Y., & Duan, D. (2024). Workshop facility layout optimization based on deep reinforcement learning. Processes, 12(1), 201. https://doi.org/10.3390/pr12010201

[34] Zhang, M., Lu, Y., Hu, Y., Amaitik, N., & Xu, Y. (2022). Dynamic scheduling method for job-shop manufacturing systems by deep reinforcement learning with proximal policy optimization. Sustainability, 14(9), 5177. https://doi.org/10.3390/su14095177

[35] Hu, H., Jia, X., He, Q., Fu, S., & Liu, K. (2020). Deep reinforcement learning based AGVs real-time scheduling with mixed rule for flexible shop floor in industry 4.0. Computers & Industrial Engineering, 149, 106749. https://doi.org/10.1016/j.cie.2020.106749

[36] Peng, Y., Lyu, Y., Zhang, J., & Chu, Y. (2025). Heterogeneous graph neural-network-based scheduling optimization for multi-product and variable-batch production in flexible job shops. Applied Sciences, 15(10), 5648. https://doi.org/10.3390/app15105648

[37] Yang, Y., & Shao, C. (2021). Hybrid multi-task learning-based response surface modeling in manufacturing. Journal of Manufacturing Systems, 59, 607-616. https://doi.org/10.1016/j.jmsy.2021.04.012