Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

SmartFactoryTwinAI: A Digital Twin and Deep Learning Framework for Virtual Production Optimization in Industry 5.0 Manufacturing Systems

abbas M. Ali mohamad

abbas.mohamad@su.edu.krd

Salahaddin University - Erbil https://orcid.org/0000-0003-3037-4903

Najdat Hameed Al-Rawi

Chiman Haydar Salh

Ali Ali Saber

Research Article

Keywords: Digital Twin, Industry 5.0, Smart Manufacturing, Deep Learning, Production Optimisation

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11568783/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

SmartFactoryTwinAI: A Digital Twin and Deep Learning Framework for Virtual Production Optimization in Industry 5.0 Manufacturing Systems

Abbas M. Ali[1], Najdat Hameed Al-Rawi[2],Chiman Haydar Salh[3], Ali Ali Saber[4]

[1,4]Alkitab university, Iraq Engineering Technical College, Department of Computers Engineering Techniques

[2]Alkitab university, Iraq Engineering Technical College, Department of Mechatronics Engineering Techniques

[3]Department of Artificial Intelligence and Robotics Engineering, Technical College of Computer and Informatics Engineering, Erbil Polytechnic University, Erbil 44001, Iraq

Author's email :abbas.mohamad@su.edu.krd

Abstract

Smart manufacturing in Industry 5.0 requires smart manufacturing systems capable of integrating virtual factory modelling, predictive analytics, adaptive scheduling, and human-centred decision support. Current approaches to digital twin and production optimisation focus on monitoring individual processes, predictive maintenance, and job-shop scheduling tasks. Through such separation, their ability to understand process-quality risks, machine reliability, exposure to downtime, and production efficiency in scheduling decisions is restricted. In response to the above, SmartFactoryTwinAI – a digital twin and deep learning-based framework for virtual production optimisations in Industry 5.0 production systems – is proposed. The proposed framework builds a tri-twin manufacturing intelligence layer, namely, Process Twin, Machine Twin and Scheduling Twin. A multilayer perceptron is used to model the risk and reliability of machine failure on the AI4I 2020 datasets, while a scheduler based on proximal policy optimisation (PPO) is used to optimise a virtual production on job-shop benchmark (JSPLib) datasets, considering reliability. Accuracies of the two models, process-quality model (95.21%) and machine failure model (97.12%), were obtained from the experimental results, and also the F1-scores were 92.45% and 94.52% for the two models; process-quality model and machine failure model were obtained from the experimental results, respectively, and ROC-AUC was found to be 95.08% and 97.06%, respectively. With the aid of JSPLib scheduling in SmartFactoryTwinAI, the makespan could be reduced to 982.41, machine utilisation increased to 93.18%, and the downtime-risk and quality-risk scores would be 0.086 and 0.097, respectively. The framework serves as a practical tool for risk-aware scheduling, what-if simulation, explainable alerts, and human-supervised smart factory decision-making.

Keywords - Digital Twin, Industry 5.0, Smart Manufacturing, Deep Learning, Production Optimisation

1. INTRODUCTION

The development of manufacturing systems is shifting from an automation-driven 'Industry 4.0' towards a human-centric, resilient, sustainable, and intelligent human-machine interaction, 'Industry 5.0 '. Digital twin technology has played a significant role in this transition, as it can be used to create a virtual production environment for the physical factory for monitoring, simulation, prediction, and decision-making [1,2]. Meanwhile, deep learning has shown promise in discovering valuable patterns from high-dimensional manufacturing sensor data, machine condition records, and production process variables [3] [4]. Hence, production optimisation, predictive maintenance, and process quality control are no longer tasks in isolation but are becoming interconnected requirements for smart factory intelligence [5, 6].

Based on the literature review, various applications of digital twins have already been developed for virtual manufacturing simulation, process monitoring, and cyber-physical production modelling, as reported in [7] and [8]. Different studies have also used machine learning and deep learning techniques for defect prediction, machine failure diagnosis and predictive maintenance in smart manufacturing environments [9] [10]. On the other hand, several heuristic rules, metaheuristic algorithms, and Adaptive algorithms based on Reinforcement learning have been studied for the job-shop scheduling and production planning [11] [12]. However, most current studies focus on these issues individually. Many digital twin investigations focus only on the representation or visualisation of a system and do not delve deeply into systems with predictive quality, intelligence, and reliability. However, in many cases, predictive maintenance models estimate the failure risk of machines and foresee production failures, but do not directly apply this knowledge in adaptive production scheduling. Also, makespan, utilisation, or idle time are often the only criteria used to optimise a scheduling solution, with no account of process-quality risk and machine-reliability variation. Such separation limits the use of current separation methods in Industry 5.0 manufacturing, where production decisions must be efficient, reliable, explainable, and appropriate for human-supervised decision-making.

The motivation for the present research is the need for a unified framework to simultaneously model process quality, machine reliability, and production scheduling within a digital twin environment. This study presents SmartFactoryTwinAI, a digital twin and deep learning framework for the virtual optimisation of production in Industry 5.0 manufacturing systems. The framework aims to predict quality-risk states from manufacturing process data, estimate machine failure risk and reliability from machine-condition data, construct a tri-twin virtual manufacturing state, and optimise the production schedule using reliability-aware deep reinforcement learning (DRL). Consequently, the study is also designed to provide explanations and scheduling justifications to keep optimised decisions understandable for production managers and engineers.

The novelty of this research is the integration of the three Twins – namely the Process Twin, the Machine Twin, and the Scheduling Twin – based on a unified tri-twin approach to operate them within a common virtual production intelligence layer. SmartFactoryTwinAI differs from traditional scheduling methods that focus on minimising completion time by also accounting for process-quality risk, machine reliability, downtime risk, and idle time, thereby establishing a multi-objective optimisation strategy. The novelty also lies in representing the outputs of predictive deep learning models as dynamic indicators for a digital twin and using them as input for reinforcement learning-based scheduling. It also provides human-centric explanations in the form of risk alerts, feature-level explanations, what-if simulation results, and schedule justifications. This makes the proposed system fit Industry 5.0 requirements regarding transparency, human supervision and responsible production decision support.

This paper makes four key contributions. The first paper presents an integrated SmartFactoryTwinAI framework that combines digital twin modelling, deep learning prediction, and DRL-based scheduling optimisation for application in smart manufacturing. Second, it builds a tri-twin representation of process quality, machine reliability, scheduling state, and production efficiency, which can be unified into one digital twin state. Third, it provides a reliability- and quality-aware production optimisation strategy that considers makespan, downtime risk, quality risk, idle time, and utilisation. Fourth, the framework is validated on public datasets and benchmarks: SECOM for process-quality prediction; AI4I 2020 for machine failure and reliability prediction; and JSPLib for job-shop scheduling optimisation. The experimental part is reproducible and methodologically up to date, with the following analyses: baseline comparison, ablation study, sensitivity analysis, what-if simulation, explainability study, and computational feasibility evaluation.

The rest of this paper is structured as follows. The literature review is provided in Section 2, and previous research on digital twins in manufacturing, predictive maintenance, process quality modelling, reinforcement learning-based scheduling, and Industry 5.0 decision support is discussed. Section 3 provides background information on digital twin-based Industry 5.0 manufacturing systems, fundamentals of deep learning, basic concepts of optimisation, and some of the notations used in the proposed framework. In Section 4, the proposed methodology, its datasets, preprocessing, the construction of the tri-twin digital twin, process-quality prediction, machine failure prediction, reliability-aware scheduling, DRL optimisation with explainable decision support, evaluation metrics, and the algorithmic workflow are described. The experimental results, in terms of predictive performance, risk analysis of the digital twin, scheduling optimisation, model comparison, ablation study, sensitivity analysis, what-if simulation, explainability, and computational efficiency, are presented in Section 5. The results, implications and limitations of the study are presented in Section 6. Finally, future research opportunities are discussed, and the paper is summarised in Section 7 for Industry 5.0 validation and real-time industrial deployment.

2. RELATED WORK

Digital twins have become more than virtual instances; they are now continuously updated, smart decision-support environments that are valuable to manufacturing. New studies involve digital twins, deep learning, reinforcement learning, simulation, edge computing, robotics, extended reality, and human-to-centre design. These developments contribute to the industry 5.0 goals of resilient production, thereby facilitating sustainable, adaptive, and human-oriented production. The current literature can be classified into five interrelated themes: Digital-twin architectures, Production optimisation, Human–Machine collaboration, AI-assisted operational intelligence, and Sustainable manufacturing ecosystems.

2.1 Digital-Twin Architectures and Industry 5.0 Integration

Manufacturing digital twins has already seen conceptual, architectural, and technological developments that address the binding between reality and its virtual representation. Most existing applications are classified by their operator, process, and product dimensions, and process-level applications have received significant attention for their potential to monitor production, detect bottlenecks, and compare configurations [1]. Digital Twin Systems Engineering takes the view a step further. It incorporates systems engineering, artificial intelligence, data fusion, and model synchronisation for ever more complex applications in manufacturing and the industrial metaverse.

In hybrid digital-twin concepts, physics-based simulation, data-driven models, domain rules, expert knowledge, and enterprise systems such as ERP, MES, and PLM are combined [6]. This kind of integration allows a twin to become more than a visualisation model; it can be used as a knowledge-based decision-support system. Empirical and review-type investigations have also demonstrated the usefulness of the DT in production planning, quality management, predictive maintenance, product lifecycle management (PLM), and operational monitoring [20]. However, interoperability, model fidelity, real-time synchronisation, security, scalability, and implementation cost are recurring issues in industrial-scale surveys [21].

Recently, smart factory architectures have been developed that fuse real-time physics-based co-simulation with edge AI and federated learning to enhance distributed intelligence and mitigate issues related to the centralisation of sensitive production data [22]. There are also lightweight digital twin solutions available for small companies and medium-sized enterprises, but the technical expertise, infrastructure costs, and integration with legacy systems remain challenges [25]. Increasingly, Industry 5.0 scenarios talk about manufacturing systems' "digital DNA", consisting of human, technological, sustainability, and resilience attributes [39]. The ability to reconfigure twins also enables changes to manufacturing requirements, such as modifications to models, resources, and control logic [47]. Automated Deployment Methods via Topology Optimisation and Multiscale Modelling aim to minimise engineering work for building and updating manufacturing twins [59]. However, such a ubiquitous architecture is still underdeveloped and fails to simultaneously ensure data synchronisation, advanced predictive modelling, virtual product testing, and closed-loop production planning.

2.2 Digital Twins for Production Planning, Scheduling, and Optimisation

Production scheduling is among the oldest-use applications of the Digital Twin. Low-volume/high-mix (LVHM) manufacturing has been used to optimise production through digital-twin technology that employs intelligent simulation and learning models to optimise job allocation, machine utilisation, order prioritisation, and mass personalisation [2]. Both flexible manufacturing twins, before being physically implemented, use discrete-event simulation and scheduling heuristics to investigate robot speed, parallel machines, dispatching policies, and workstation configuration [10].

The adaptability of production twins is strengthened by reinforcement learning. Digital twins based on Soft Actor–Critic can adapt to fluctuations in demand, uncertainty, and changes in resource availability, thereby enhancing productivity, utilisation, energy efficiency, and system flexibility [14]. Multilevel production-planning models also link strategic, tactical and operational decisions and include human-oriented, sustainable and resilient Industry 5.0 requirements [16]. These features have been further expanded by adaptive planning platforms that use AI to enable continuous sensing, prediction, reliability assessment, and dynamic adjustments to production plans [27].

Assembly-station design with multi-objective optimisation has considered productivity, ergonomics, resource utilisation, and worker-related constraints together using virtual simulation [28]. Self-learning digital twins for industrial robots are employed to continuously update decision policies and make regular adjustments to robotic actions based on operational feedback, to mitigate variations in manufacturing conditions [30]. The semantic representation of machines, jobs, constraints, and process relationships is provided by knowledge-graph-enhanced twins, which also enhance job-shop scheduling and explainability [38]. An evolutionary optimisation of scheduling based on statistical learning has also been proposed for digital-twin-informed scheduling under discrete and dynamic production conditions [44]. A CNC digital twin facilitates real-time monitoring of a machine's operation, visualisation of the machining state, CNC fault detection, and process supervision at the

machine level. While some of these methods maximise the optimality of a scheduling objective or task, several do not consider production throughput, energy use, equipment degradation, product quality, and human workload.

2.3 Human-Centric Digital Twins and Collaborative Manufacturing

To distinguish the industry 5.0 twins from largely automation-focused Industry 4.0 systems, human-centricity is introduced. Human digital-twin frameworks involve human operators in the cyber–physical feedback loop, enabling optimised interaction, safety monitoring, and skill development [3]. Human-robot teaming, machine learning, digital twins, advanced sensing, and mixed reality (MR) interfaces show potential for collaborative perception, prediction, and decision-making [7]. Physiological, behavioural, ergonomic and cognitive aspects can be simulated in human digital twins, but there is a lack of both operational approaches and standardised evaluation measures [8].

Multirobot systems based on digital twins include motion planning, task distribution, communication estimation, vision-based servoing, and human-in-the-loop considerations and supervision [9]. Four-dimensional path-planning twins also account for spatial trajectories, time, robot velocity, and static and dynamic obstacles to coordinate the safe operation of collaborative robots [12]. Conceptual studies highlight that digital twins are not meant to replace workers but to enhance their skills and creativity in decision-making [18], and reviews recognise the key technologies enabling digital twin applications, such as wearable sensing, artificial intelligence, simulation, and interfaces [19].

Interoperable robotic architectures are designed to enable bidirectional data exchange between industrial digital-twin platforms and robotic middleware, enabling flexible human–robot collaboration modes [31]. The quasi-personalised assembly, in which workers' capabilities, ergonomics, workstation configuration, and production requirements are jointly considered, is one area where human-centred twins have been applied. Digital twins are applied to reconfiguration frameworks that synchronously orchestrate the reconfiguration of human and robot resources in a cyber–physical production system [35]. The worker-performance twins assess Human–Machine Interaction based on task execution, physical effort, operational precision, and performance indicators [36]. Twins, based on Augmented Reality (AR), enhance situational awareness by providing contextual manufacturing information within the operator's workspace [37].

Use of a deep-learning-enhanced twin can predict unsafe human–robot interactions and enhance the collaborative system's reliability [42]. Extended-reality transfer methods are used to immerse the user in immersive interaction with virtual production models [49], and robot-manipulation twins are used to validate trajectories and control operations safely before deployment [51]. Cross-belt sorting and logistics systems have also been integrated with a human–machine twin [53]. Even with these advancements, relatively few studies integrate all four components within a single closed-loop framework: worker-state prediction, machine behaviour, production objectives, and explainable optimisation.

2.4 Deep Learning, Intelligent Control, Maintenance, and Cybersecurity

Artificial intelligence augments digital twins, translating real-time production data into predictions and control actions. Applications in various sectors illustrate how the twin can be used to estimate equipment condition, increase test transparency and support safer manufacturing processes without added cost [5]. Embedding Vision Transformer enables anomaly detection, quality inspection, maintenance, and adaptive optimisation by analysing complex visual and operational data [11]. Robotic systems, production line control, remote monitoring, and intelligent optimisation are further connected with the IoT--cloud-connected production twins [15].

Lifecycle-oriented industrial metaverse systems utilise AI and digital twins to manage machine design, operation, maintenance, and service information [23]. Two forms of personalisation are introduced via the twins: AI-based models for individual products and individual decision strategies for the worker and the production scene. Product statement-updated: twins have also been applied in electronic manufacturing systems for intelligent signal control [40], and, increasingly, robots are taking on sustainable product creation, autonomous control, and flexible manufacturing. Digital Twins and AI are suggested in lean-manufacturing roadmaps as a means to decrease waste, increase responsiveness and enable continual improvement [43].

Product design and virtual commissioning of these systems enable CAx-integrated methodologies to enhance the link among design, engineering analysis and digital-twin validation. This has been extended to the design evaluation, performance prediction, and lifecycle optimisation of agricultural machinery using data-driven models supported by AI technology [48]. Digital twins for cybersecurity are designed to simulate cyber–physical production environments, enabling evaluation of vulnerabilities and testing of cyber defence strategies without impacting physical production environments [50]. Nevertheless, secure real-time twins for AR/VR-enabled factories face risks related to latency, authentication, communication integrity, privacy, and synchronisation [52].

5.0 maintenance takes into account AI, resilience analysis, sustainability, and human input for proactive asset-management decisions [54]. Other digital twins, such as graph-based representations and machine learning techniques using label encoding, have also been used in SCADA to identify cyber attacks [58]. A trusted, integrated intelligence layer is required, as existing systems generally focus on prediction, optimisation or security as separate functions.

Table 1 provides a thematic overview of available digital-twin and AI studies related to manufacturing, synthesising the main research directions, technical contributions, and open challenges in the literature.

Table 1: Thematic Summary of Existing Research on Digital Twins and AI for Industry 5.0 Manufacturing

S. No.	Research Theme	Primary Focus	Common Technologies and Methods	Major Contributions	Identified Research Gap	Representative References
1	Digital-twin architectures and system integration	Development of synchronised virtual representations for products, processes, operators, and manufacturing assets	IoT, cyber–physical systems, ERP–MES–PLM integration, physics-based simulation, hybrid knowledge models, edge computing, federated learning	Established architectural foundations for real-time monitoring, simulation, lifecycle management, and enterprise-level decision support	Limited interoperability, scalability, semantic consistency, model synchronisation, and integration with legacy manufacturing systems	[1], [4], [6], [20]–[22], [25], [39], [47], [59]
2	Production planning, scheduling, and virtual optimisation	Optimisation of jobs, machines, production lines, robotic operations, and resource allocation	Discrete-event simulation, scheduling heuristics, reinforcement learning, multi-objective optimisation, knowledge graphs, evolutionary algorithms	Improved makespan, resource utilisation, adaptability, production flexibility, and response to demand variations	Most approaches optimise isolated objectives and inadequately consider energy, quality, equipment degradation, uncertainty, and worker constraints simultaneously	[2], [10], [14], [16], [27], [28], [30], [38], [44], [57]
3	Human-centric digital twins and human–robot collaboration	Integration of workers, collaborative robots, ergonomics, safety, and human decision-making into digital-twin environments	Human digital twins, wearable sensing, AR/MR/XR, collaborative robotics, path planning, behavioural modelling, human-in-the-loop control	Enhanced worker safety, personalised assistance, ergonomic assessment, collaborative task planning, and operational awareness	Lack of unified models combining worker state, machine behaviour, production performance, explainability, and real-time adaptation	[3], [7]–[9], [12], [18], [19], [31], [34]–[37], [42], [49], [51], [53]
4	Deep learning, predictive intelligence, and	Prediction of equipment states, defects, anomalies, production conditions,	CNNs, Vision Transformers, deep neural networks, reinforcement learning, signal	Enabled predictive maintenance, visual inspection, anomaly	Limited generalisation across factories, insufficient uncertainty	[5], [11], [15], [23], [24], [40], [41], [43], [46], [48], [54]

	autonomous control	and optimal control actions	processing, intelligent agents, IoT–cloud analytics	detection, adaptive control, condition monitoring, and data-driven decision support	estimation, weak explainability, and inadequate closed-loop learning from physical feedback	
5	Cybersecurity, reliability, and trustworthy digital twins	Protection of connected digital-twin infrastructures and reliable operation of cyber–physical manufacturing systems	Cyber-range simulation, graph learning, SCADA attack detection, secure communication, authentication, privacy mechanisms, reliability analysis	Supported vulnerability assessment, cyberattack detection, virtual security testing, and safer connected manufacturing operations	Security is generally treated separately from production optimisation, while privacy, trust, adversarial robustness, and secure synchronisation remain underexplored	[42], [50], [52], [54], [58]
6	Sustainable, resilient, circular, and immersive manufacturing	Alignment of digital twins with sustainability, resilience, workforce training, circularity, logistics, and industrial metaverse applications	Lifecycle simulation, circular manufacturing models, virtual learning environments, industrial metaverse, immersive simulation, collaborative platforms	Improved resource efficiency, waste reduction, workforce development, lifecycle visibility, resilience, and distributed collaboration	Existing studies lack an integrated multi-objective framework that jointly balances productivity, energy efficiency, resilience, sustainability, and human well-being	[13], [17], [26], [29], [32], [33], [45], [55], [56], [60]

Table 1 reveals that the existing research has developed digital-twin architecture, scheduling, human–robot collaboration, predictive intelligence, cybersecurity, and sustainable manufacturing. The literature, however, remains scattered and application-specific. Not many frameworks include techniques for continuous physical–virtual synchronisation, state prediction using deep learning, virtual scenario evaluation (VSE), decision-making under uncertainty, and multi-objective production optimisation. SmartFactoryTwinAI aims to bridge this gap by creating a single digital twin deep learning framework to optimise production reliably, sustainably, adaptively, and in a human-centric way in Industry 5.0 manufacturing systems.

2.5 Sustainable, Resilient, and Immersive Manufacturing Ecosystems

Digital twins are becoming integral to sustainability, extending beyond individual production machines. The main application of the twin-based virtual learning environments (VLEs) is an immersive learning platform that offers training, adaptive learning, remote experimentation, and workforce skills development in the context of smart manufacturing [13]. The platforms of the industrial metaverse are an integrated system of digital twins, a collaborative environment, simulation and immersive interfaces for sustainable production planning and distributed engineering [17]. Intelligent manufacturing, human–machine cooperation, resilience, sustainability, and personalisation are intertwined and recognised as research directions in Industry 5.0, as noted in concept reviews that span appraisals by various stakeholders throughout the entire manufacturing process [26].

The concept of digital transformation frameworks links performance optimisation to environmental and social sustainability at the supply chain level [29]. The twin in the circular manufacturing facilitates resource recovery, remanufacturing, lifecycle tracking and closed-loop material flows [32]. In the shipping industry, digital twins have been deployed to monitor assets in real time, coordinate routes, manage disruptions, and provide operational

resilience as well [33]. Human-centric and sustainable production decisions can be made using decentralised structures as holonic, flexible, and reconfigurable manufacturing concepts [45]. A further advantage of Metaverse-enabled manufacturing operations is collaborative planning, virtual validation, and lifecycle visibility [55], as well as integrated three-dimensional simulation and digital-twin environments, which help with collaborative production-process design and planning [56]. Creating an AI-ready workforce for AI-manufacturing systems by preparing them for digital twin-based teaching and learning is also a high priority.

Overall, previous research has shown progress in various aspects, including simulation, scheduling, human–robot collaboration, maintenance, security, sustainability, and immersive interaction. However, the solutions are mostly application-specific, rely on reducing specific targets or do not continuously learn from two-way production data. This drives SmartFactoryTwinAI - a synchronised manufacturing digital twin and deep information-based state predictions, virtual scenario evaluation, and multi-objective production optimisation. The proposed framework is designed to optimise productivity, resource and energy usage, equipment reliability, and human-centric operational constraints within the adaptive Industry 5.0 smart-factory environment.

3. PRELIMINARIES

The basic concepts needed to understand the proposed SmartFactoryTwinAI framework are introduced in this section. It explores the concepts of digital twin in Industry 5.0 manufacturing, cyber-physical production modelling, deep learning in production process intelligence, machine reliability prediction and production optimisation. These preliminaries lay the theoretical groundwork for embedding virtual factory simulation, predictive analytics and human-centric decision support into intelligent manufacturing systems.

3.1 Digital Twin-Enabled Industry 5.0 Manufacturing Systems

In manufacturing, a digital twin is a virtual version of a physical production system that is dynamically updated to mirror the behaviour of the production system, process conditions, machine conditions and production results within the real-world factory environment. The difference with a conventional simulation model, which is typically static, and a digital twin is that the digital twin should be capable of providing dynamic monitoring, prediction, and decision support by aligning the physical manufacturing data with the virtual model. In a smart manufacturing system, this synchronisation is facilitated via the cyber-physical production system, which integrates sensors, machines, controllers, industrial networks, and computational models to capture and represent the behaviour of the factory in real time.

Industry 5.0 is an evolution of Industry 4.0, focusing on human-centricity, sustainability, resilience, and collaborative intelligence between humans and intelligent machines. Digital twins are not just for production visualization or production reproduction; they can be intelligent environments for decision support where production managers and production engineers can simulate quality risks, machine reliability, scheduling options, and production efficiency before acting upon decisions in the real shop floor. Thus, the digital twin serves as a link between data intelligence and human-assisted production control.

The manufacturing state in the digital twin can be described as:

$$DT_t = \{Q_t, R_t, S_t, E_t\}, \tag{1}$$

where DT_t is the state of the digital twin at time t, Q_t is the process quality state, R_t is the machine reliability state, S_t is the production scheduling state, and E_t is the production efficiency state. However, as illustrated in equation (1), it is not enough to model a digital twin-enabled manufacturing system merely through the variables of the machines or processes. Rather, it should be a combination of factors of quality, reliability, scheduling and efficiency, to support practical production optimization.

The idea described in Eq. (1) was adopted in SmartFactoryTwinAI as the theoretical foundation of building an intelligent virtual production environment. The framework does not view the digital twin as a fixed version of the factory; instead, the digital twin is a layer of factory information and intelligence that is constantly updated, with process quality, machine health, scheduling decisions and production efficiency all modelled at the same time. Thereby, the framework is appropriate for Industry 5.0 manufacturing, where automated optimization needs to be explainable, reliable, sustainable, and in support of human decision-makers.

3.2 Deep Learning and Optimization Foundations for Smart Manufacturing

The high dimensionality, nonlinearity, and heterogeneity of the data collected from smart manufacturing systems in the form of sensors, machines, controllers, and production monitoring systems make deep learning a crucial

modelling technique. With these data streams, process quality prediction, abnormal production behaviour detection, machine degradation estimation and operational decision making can be performed. In the context of digital twin-enabled manufacturing, deep learning models serve as predictive intelligence modules that decode raw production data to feed virtual factory analysis with actionable states.

Typically, process quality prediction is a supervised learning problem where the manufacturing sensor variables are mapped to some quality state or probability of a defect. Suppose that X_p is the process feature matrix from manufacturing sensor measurements. The state of process quality can be expressed as:

$$\hat{Q}_t = f_\theta(X_p), \tag{2}$$

here, f_θ denotes the process-quality prediction model with learnable parameters θ; $\hat{Q}_t$ is the predicted quality state at time t. The basis for determining production states that might result in defective outputs or unstable manufacturing behaviour is given in Eq. (2).

Machine failure and reliability prediction are also vital in smart manufacturing since machine degradation clearly impacts downtime, throughput, production cost and schedule stability. Assume the machine-condition feature matrix X_m can be derived by the machine operating variables, including temperature, rotational speed, torque and tool wear. The machine failure-risk prediction can be formulated as:

$$\hat{R}_t = g_\phi(X_m), \tag{3}$$

where g_ϕ denotes the machine-health prediction model with learnable parameters ϕ, and $\hat{R}_t$ represents the predicted machine failure risk at time t. The machine reliability score is based on this risk estimate and defined as:

$$Rel_t = 1 - \hat{R}_t, \tag{4}$$

where Rel_t is the estimated reliability of the machine at time t. Predictive maintenance intelligence can be transformed into predictive reliability information for production decisions as displayed in Eq. (3) and (4).

The production scheduling problem in manufacturing can often be formulated as a job-shop scheduling problem, i.e., a set of jobs has to be processed on a set of operating machines with sensitivity to the order of their processing and subject to some machine capacity restrictions. Conventional scheduling techniques are primarily concerned with minimization of makespan or completion time. However, for smart manufacturing, the makespan of a resulting schedule may be not enough since a lower makespan schedule might be unstable due to the high failure risk of the machines assigned to the critical jobs or neglect of quality-related instability. Thus, multi-objective production optimization is needed to optimize productivity, reliability, quality and resource utilization.

The overall multi-objective optimization can be written as:

$$\min Z = \alpha C_{max} + \beta D_r + \gamma Q_r + \delta I_t, \tag{5}$$

In the above, C_{max} is the makespan, D_r is the downtime risk, Q_r is the quality risk, I_t is the idle time and α, β, γ, and δ are objective weights. Eq. (5) shows that optimizing production should not rely solely on time based performance but reliability-aware and quality-aware factors should also be taken into account.

The adaptive production scheduling problem is suitable to be addressed using deep reinforcement learning, which allows an agent to learn scheduling decisions by repeatedly interacting with a virtual production environment. In this context, the agent takes in the current manufacturing condition, performs a scheduling action, and gets a reward from the production performance, and adjusts its scheduling policy based on the reward. This is a learning process by rewards, and very convenient in the context of digital twins, where various production scenarios can be tested virtually before being implemented. The foundations, defined in Equations (2)-(5), enable the integration of predictive modelling and optimization in SmartFactoryTwinAI, and process-quality and machine-reliability estimates are leveraged to create more efficient and reliable production schedules.

Table 2: List of Notations Used in SmartFactoryTwinAI Framework

Symbol(s)	**Description**
DT_t	Digital twin state at time t
$Q_t, \hat{Q}_t, Q_r$	Process quality state, predicted quality-risk score, and accumulated quality risk

$R_t, \hat{R}_t, Rel_t$	Machine reliability state, predicted failure-risk score, and reliability score
S_t, E_t	Production scheduling state and production efficiency state
D_t, D_r, I_t	Downtime-risk state/penalty, accumulated downtime risk, and idle-time state/penalty
X_p, X_m	Process feature matrix from SECOM and machine-condition feature matrix from AI4I 2020
x_i, x_i', μ, σ	Original feature value, normalized feature value, feature mean, and standard deviation
f_θ, θ	Process-quality prediction model and its learnable parameters
g_ϕ, ϕ	Machine failure-risk prediction model and its learnable parameters
L_q, y	Process-quality prediction loss and actual class label
S, S^*	Job-shop scheduling representation and optimized production schedule
J, M, O, P, C	Set of jobs, machines, operations, processing-time matrix, and precedence constraints
s_t, a_t, r_t	DRL state, selected scheduling action, and reward at time t
j, m	Selected job/operation and selected machine
C_t, C_i, C_{max}	Completion-time penalty, completion time of job i, and makespan
Z	Multi-objective optimization value
$\alpha, \beta, \gamma, \delta$	Objective weights for makespan, downtime risk, quality risk, and idle time
$\lambda_1, \lambda_2, \lambda_3, \lambda_4$	Reward weights in the DRL reward function
$BusyTime_m, AvailableTime_m$	Active processing time and total available time of machine m
TP, TN, FP, FN	True positives, true negatives, false positives, and false negatives
E_p	Number of DRL training episodes

Table 2 summarizes the main mathematical notations used in SmartFactoryTwinAI, covering digital twin states, predictive models, scheduling variables, optimization weights, and evaluation terms.

4. MATERIALS AND METHODS

The proposed SmartFactoryTwinAI framework is explained in this section together with the public datasets, the applied preprocessing, the creation of the digital twin, the deep learning prediction modules, and the proposed reliability-aware production optimization process. It sheds light on how process quality, machine reliability and scheduling states are combined together in a virtual manufacturing environment to enable adaptive production decision making and enhance explainability and humanisation of Industry 5.0 systems.

4.1 Overview of the Proposed SmartFactoryTwinAI Framework

The SmartFactoryTwinAI framework is a comprehensive digital twin-based production optimization system for an Industry 5.0 factory based on deep learning. With process monitoring, machine-health prediction and virtual job-shop scheduling, the framework integrates the three into a single manufacturing intelligence pipeline. The ultimate aim is to provide and facilitate time-efficient production decisions which are also reliability-aware, quality-aware and human-supervised in industrial decision making.

SmartFactoryTwinAI is built based on three public datasets, which are representing three functional aspects of smart manufacturing. Process-state behaviour and prediction of quality-related production risks are modelled based on the SECOM Semiconductor Manufacturing Dataset. The AI4I 2020 Predictive Maintenance Dataset can be used to predict the risk of machine failure and reliability conditions. The JSPLib Job Shop Scheduling Benchmark is employed to generate virtual production scheduling scenarios consisting of jobs, machines,

operation sequences and processing times. This merging of the three data sources provides a wider virtual manufacturing environment, in which quality, reliability and scheduling can be analysed jointly.

SmartFactoryTwinAI comprises six key steps. Firstly, public manufacturing datasets are gathered and mapped to process, machine and scheduling streams. Second, dataset specific pre-processing is done to address missing values, normalize numerical data, encode categorical data and convert scheduling problems to structured job-machine representations. Third, a digital twin layer is built to model the virtual factory state with process quality, machine reliability, scheduling condition and production efficiency indicators. Fourth, deep learning modules are used to predict process quality risk and machine failure risk based on the manufacturing data. Fifth, the predicted quality and reliability scores are provided to a production optimization layer, which will consider the reliability scores. Finally, the optimized schedules and risk alerts, utilization indicators, and explainable decision outputs are presented for human-in-the-loop Industry 5.0 decision making.

The proposed framework is based on the relationship between the digital twin, deep learning models and deep reinforcement learning scheduler. The virtual manufacturing environment is represented by the digital twin, and the deep learning modules create predictive intelligence on process stability and machine reliability. These predicted states are then utilized by the deep reinforcement learning scheduler to make job-machine allocation decisions and enhance production performance through virtual interaction. This design enables the system to try out production decisions in a simulation environment before implementing them in real production.

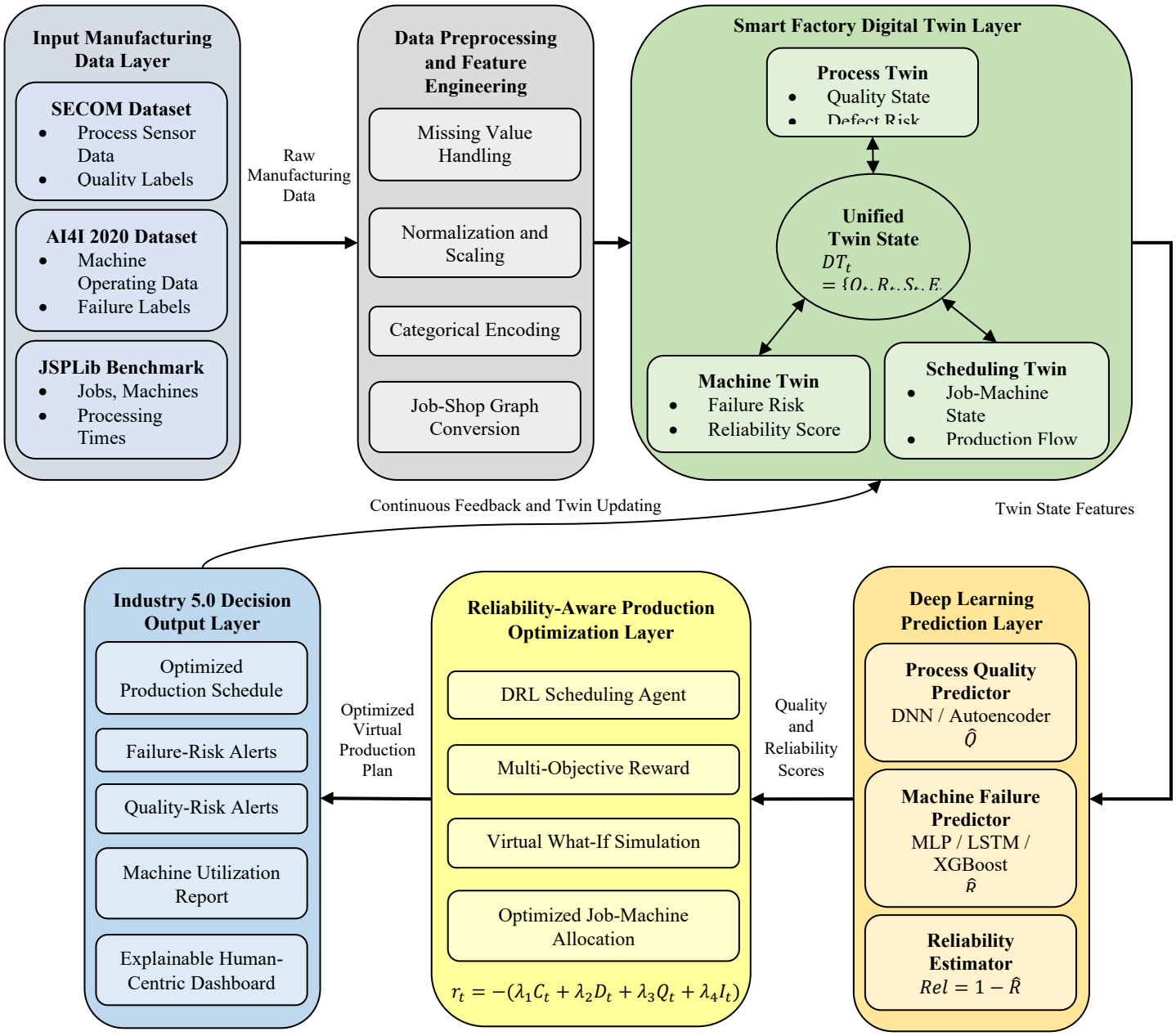


Figure 1: Overall Architecture of SmartFactoryTwinAI for Digital Twin-Driven Industry 5.0 Manufacturing Optimization

The overall architecture of SmartFactoryTwinAI is given in Figure 1. It illustrates the flow from SECOM, AI4I 2020 and JSPLib inputs, to preprocessing, digital twin construction, deep learning-based prediction, reliability aware DRL optimization and Industry 5.0 decision outputs. The figure also shows the seamless feedback loop between simulated results and human inspection that can be used to change the virtual production state. In this way, SmartFactoryTwinAI offers an end-to-end methodology to integrate predictive analytics, digital twin modelling and adaptive production optimisation in smart manufacturing systems.

4.2 Public Datasets and Experimental Materials

These experimental materials are chosen and represent the three key aspects of SmartFactoryTwinAI: process quality modelling, machine-health prediction, and virtual production scheduling. Given the lack of accessibility of real factory level digital twin datasets because of industrial confidentiality, publicly available and reproducible datasets are used here to include signals with sufficient information regarding manufacturing to validate the proposed framework. To model the process sensor behaviour and the product-quality outcome, the SECOM Semiconductor Manufacturing Dataset is used. The AI4I 2020 Predictive Maintenance Dataset is used to model the machine operating conditions and the risk of failure. The Job Shop Scheduling Benchmark, called JSPLib, is utilized to develop a virtual production scheduling scenario in terms of job-machine allocation and optimization.

The SECOM data set is an example of semiconductor manufacturing measurements that are taken from real semiconductor manufacturing process with high dimensional sensor variables and pass/fail quality labels. In SmartFactoryTwinAI, this dataset is utilized to learn and test the process-quality prediction part. The AI4I 2020 data set includes synthetic yet industry-relevant PM data with information regarding operating variables of the machine and failure labels. It is used to calculate risk of failure and obtain the machine reliability indicators for the digital twin. JSPLib is a set of standard job-shop scheduling instances that include jobs, machines, sequences of the operations, and processing times. It's used for testing the production optimization layer in controlled and reproducible scheduling scenarios.

Table 3 summarises the datasets which are used for the validation of SmartFactoryTwinAI. The distinct methodological roles are given to each dataset to prevent overlap between process modelling, machine reliability estimation and production scheduling evaluation.

Table 3: Summary of Public Datasets Used for SmartFactoryTwinAI Validation

Dataset	Dataset Characteristics	Purpose in Framework	Task	URL	Citation
SECOM Semiconductor Manufacturing Dataset	Semiconductor process dataset with high-dimensional sensor measurements and pass/fail quality labels	Process-state and quality modelling	Defect prediction and process-quality risk estimation	https://archive.ics.uci.edu/dataset/179/secom	[61]
AI4I 2020 Predictive Maintenance Dataset	Industrial predictive maintenance dataset with machine operating features, failure labels, and failure-type information	Machine-health and failure-risk modelling	Predictive maintenance and reliability-score estimation	https://archive.ics.uci.edu/dataset/601/ai4i+2020+predictive+maintenance+dataset	[62]
JSPLib Job Shop Scheduling Benchmark	Standard job-shop benchmark instances with jobs, machines, operation order, and processing times	Virtual production scheduling and optimization	Job-shop optimization and makespan minimization	https://scheduleopt.github.io/benchmarks/jsplib/	[63]

All three of the datasets are applied in complementary phases of the proposed methodology. SECOM helps the process twin by allowing prediction of quality-risk based on manufacturing sensor variables. AI4I 2020 offers

failure-risk and reliability information to support the machine twin. The scheduling twin is supported by JSPLib, which includes structured production instances for the evaluation of job-shop optimization. These datasets enable the validation of SmartFactoryTwinAI as a reproducible digital twin-based framework for quality-aware, reliability-aware production optimization for efficiency.

4.3 Data Preprocessing and Feature Engineering

The three public datasets are preprocessed to maintain their functionality in the SmartFactoryTwinAI framework. The SECOM, AI4I 2020, and JSPLib datasets are structured differently, therefore different preprocessing pipelines are used before being mapped into the digital twin environment. SECOM is transformed to be a process-quality prediction problem, AI4I 2020 is transformed to be a machine failure-risk estimation problem, and JSPLib is transformed to be a structured problem of job-machine scheduling.

The SECOM dataset has the property that some of the semiconductor process variables might have missing measurements, which are first imputed using the median imputation. Too many missing values or values that are too close to each other are deleted as this causes noise and can lead to unstable model training. Remaining numerical sensor variables are normalized after imputation to make sure that all process features contribute on a similar scale. The normalization of the feature value is calculated as:

$$x_i' = \frac{x_i - \mu}{\sigma}, \tag{6}$$

where x_i is the original feature value, μ the feature mean, σ is the feature standard deviation and x_i' is the normalized value. The numerical process and machine-condition features used by the predictive models are applied to Eq. (6).

In the case of AI4I 2020 dataset, the numerical values like air temp, process temp, rotational speed, torque, and tool wear are normalized using Equation (6). One-hot encoding is used to convert the product-type categorical attribute to numerical. The machine failure label is created as the primary binary class for failure-risk prediction and available failure-type labels can also be kept for auxiliary multi-class prediction if desired. This preparation helps the machine-health model to predict failure probability and reliability information from operating conditions.

The JSPLib benchmark consists of structured job-machine-operation representation of the scheduling instances. A job is described as a sequence of operations, each of which can be done on a particular machine with a known processing time. Scheduling representation is given as:

$$S = \{J, M, O, P, C\}, \tag{7}$$

where J denotes the set of jobs, M denotes the set of machines, O denotes the set of operations, P denotes the processing-time matrix and C denotes the precedence constraints among operations. The structured representation given by Eq. (7) can be used in the construction of the virtual job-shop scheduling environment.

Predictive datasets are split into training, validation and test data sets, stratifying the split to ensure class distribution is maintained, especially when the number of defective samples is less than the number of normal samples and the number of failure events is less than the number of successful events. In the development of the model and its assessment, a common split of 70% training, 15% validation and 15% testing is adopted. Since class imbalance may be observed, "balancing" techniques, like class weighting, random oversampling, or synthetic minority oversampling may be applied only to the training set to prevent information leakage. The processed outputs of SECOM, AI4I 2020 and JSPLib are then provided to the process twin, machine twin, and scheduling twin respectively.

4.4 Smart Factory Digital Twin Construction

The smart factory digital twin is built as a virtual manufacturing environment, which brings together process quality, machine reliability, scheduling status and production efficiency together in a unified operational status. Unlike a simple visualization of the production system, the digital twin in SmartFactoryTwinAI is also an intelligent simulation and decision-support layer, which is fed with processed manufacturing data, updates the virtual factory states and provides optimization-ready data for the production scheduler.

The proposed digital twin consists of three subtwins, namely, the Process Twin, the Machine Twin and the Scheduling Twin. In the case of the manufacturing process, the Process Twin is an indicator of the quality-related behaviour of the manufacturing process, based on process features measured by the sensors. It is used to make estimates of how the process is currently operating and to determine if a process state is likely to produce a defective product. The Machine Twin is a representation of the operational health of machines, including

modelling of failure risk, reliability level and potential downtime behaviour. The Scheduling Twin stores information on jobs, machines, sequence of operations, processing times, machine assignment and machine idle times, representing production flow.

The unified digital twin state is represented as in Equation (1) but, the virtual factory state at time tis is represented by process quality, machine reliability, scheduling condition, and production efficiency. It is proposed that in the implementation, Q_t is obtained from the process-quality modelling stream, R_t is obtained from the machine-health prediction stream, S_t is received from the job-shop scheduling stream, and E_t is computed from production indicators, such as machine utilization, idle time, and completion status. This single representation enables the framework to consider production decisions based on more than a single scheduling objective.

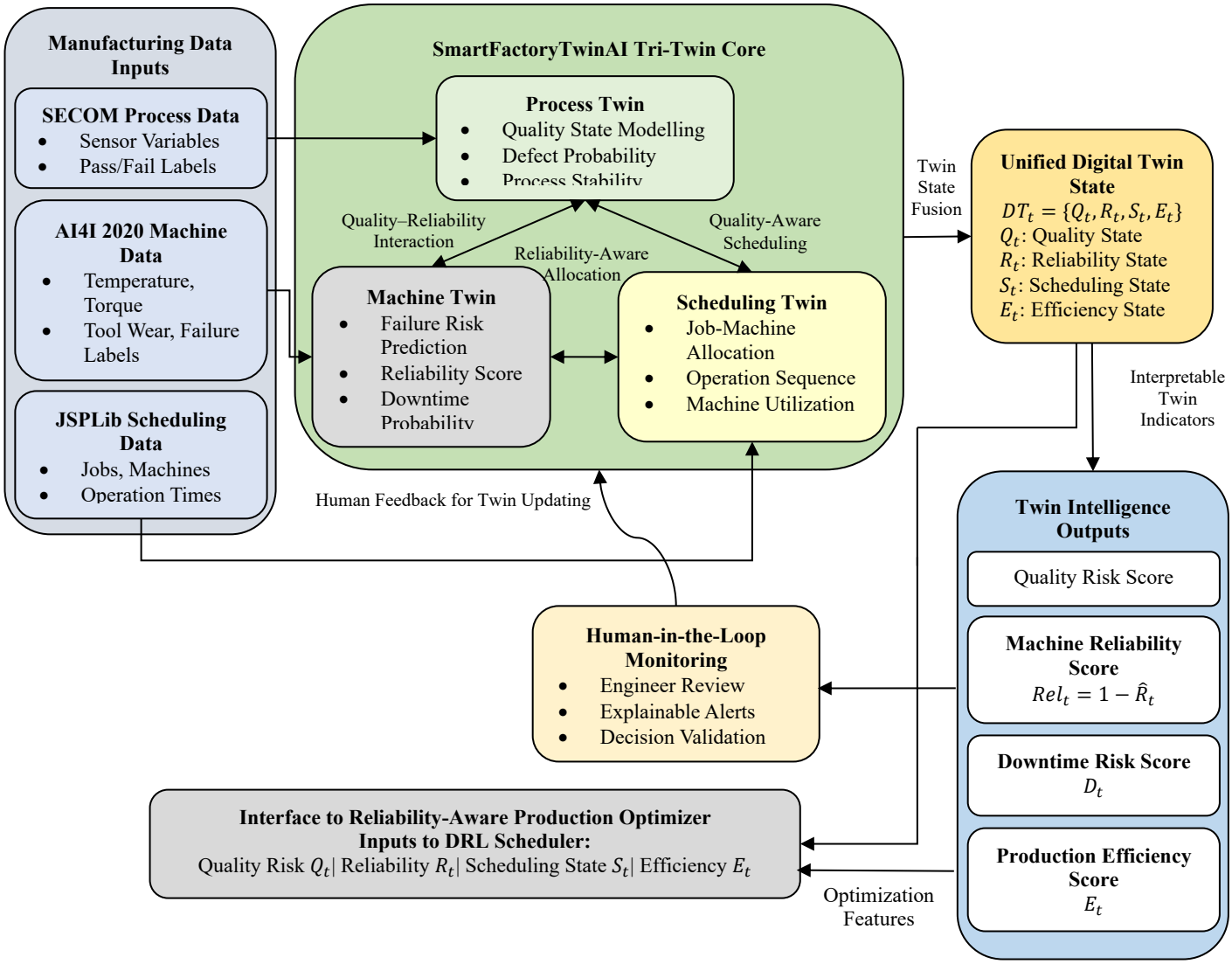


Figure 2: Tri-Twin Manufacturing Intelligence Layer for Process Quality, Machine Reliability, and Scheduling State Modelling

Whenever new predictive information or scheduling information is available, twin state updating is performed. When the process model detects a higher quality risk, the Process Twin updates Q_t. If the machine-health model indicates a change in the probability of failure or reliability, then the Machine Twin updates R_t. Every scheduling action has an impact on the Scheduling Twin which is updated as per the schedule with respect to operation completion, machine load, and remaining job status. Then, the efficiency state E_t is updated with the last scheduling and utilization information. This update mechanism allows to synchronize both the physical production data and the virtual manufacturing state continuously.

Data-to-twin mapping maps a relationship between the physical production data and the virtual production state. Process measurements from SECOM are mapped to the Process Twin, machine operating variables from AI4I 2020 are mapped to the Machine Twin and job-machine-operation structures from JSPLib are mapped to Scheduling Twin. These are not live streams from the factory but experimental resources that are publicly

available, which enable the reproduction of key manufacturing dimensions that are necessary to validate the proposed digital twin logic.

The tri-twin manufacturing intelligence layer of SmartFactoryTwinAI is shown in Figure 2. It illustrates the inputs to Process Twin, Machine Twin, and Scheduling Twin and the inputs that they provide to the unified digital twin state. It also shows the quality risk, reliability information, downtime risk and scheduling status transfer to the production optimization module. In this way, the digital twin created will be the primary virtual space used to link the predictive modelling and reliability-aware production optimization with SmartFactoryTwinAI.

4.5 Process Quality Prediction Module

The process quality prediction module indicates if a manufacturing process state is normal or defective. For SmartFactoryTwinAI, this module is trained with the SECOM Semiconductor Manufacturing Dataset, which is high dimensional process sensor measurements with a quality label (pass/fail). This module primarily aims to convert the process sensor characteristics into a quality-risk indicator, which can then be used by the Process Twin and subsequently given to the production optimization layer.

After missing value imputation, feature filtering, and normalization by SECOM, the preprocessed process feature matrix is denoted as X_p. The quality prediction task is a supervised mapping similar to that defined in Equation (2), with the process-quality model f_θ predicting the quality state $\hat{Q}_t$ from X_p. The f_θ can be realized with a deep neural network, a hybrid multilayer perceptron or an autoencoder-classifier structure in this study. The autoencoder part can learn a low dimensional representation of the high dimensional process variables and the classifier part can be used for predicting the final quality state.

The module's output is a binary process condition: normal class, corresponding to acceptable production behaviour and defective class, corresponding to a quality-risk state. Along with the last class label, the predicted probability of the defective class is considered as quality-risk score. This score enables the digital twin to capture process instability continuously, not just by making a binary normal/defective decision.

The training of the process-quality model is done using the binary cross-entropy loss function defined as:

$$L_q = -[y\log(\hat{Q}_t) + (1-y)\log(1-\hat{Q}_t)], \tag{8}$$

where L_q is the process-quality prediction loss function, y is the real quality label and $\hat{Q}_t$ is the output of the process-quality from the model in Equation (2). The loss in Eq. (8) encourages the model to make correct predictions by raising the loss if the prediction is not the same as the true label.

The predicted quality output is used as the process quality state Q_t after training and passed to the Process Twin. The probability of a defective class is related to the risk of production quality: the larger the probability, the greater the risk of production quality is poor, and the smaller the probability, the more stable the production condition is. This quality-risk information is used by SmartFactoryTwinAI to take not only completion time into account when deciding on scheduling, but also its likely impact on production quality.

The Machine Failure and Reliability Prediction Module is the closest to the actual application of the program.

The machine failure and reliability prediction module predicts the risk of machine operation before generating the scheduling decisions. The module in SmartFactoryTwinAI is based on the AI4I 2020 Predictive Maintenance Dataset, which offers machine operating conditions and failure related labels. This module is for transforming raw machine-condition data into failure probability, reliability-score, and downtime-risk data for use by the Machine Twin and the reliability-aware production optimizer.

The input feature matrix X_m includes air temperature, process temperature, rotational speed, torque, tool wear and product type. The numerical features are normalized as mentioned in Section 4.3 and the product-type variable is transformed to a numerical representation prior to the training of the model. There are two ways of setting the target variable. Firstly, binary failure prediction is conducted and machine failure label is used to classify whether the machine will fail or remain operational. Second, based on the experimental setting, the multi-class failure prediction can be used to classify the different failure types, such as tool wear failure, heat dissipation failure, power failure, overstrain failure and random failure.

The machine-health prediction task is modeled in the same manner as defined in Equation (3) above, but for a mapping from the machine-condition feature matrix X_m to the estimated failure risk $\hat{R}_t$. The model can be realized by Multilayer Perceptron, Gradient Boosting Classifier or Hybrid Deep Learning Classifier based on the comparative experimental design. The failure probability that is predicted is considered as the machine risk

indicator, and the reliability score is calculated by using the equation (4). Thus, a high failure probability will yield a low reliability score and a low failure probability will indicate a more stable machine condition.

The reliability score that is generated in this module is added as the current machine reliability state R_t to the Machine Twin. This value is used to determine suitability of each machine for future job assignments. Those machines that have lower reliability scores are more prone to causing production disruptions, late completion, or erratic scheduling results. Hence, the reliability state is not merely a maintenance signal, but also as an optimization signal in production scheduling.

The risk of downtime is calculated from the failure probability predicted and from the impact of failure by assigning jobs to a machine under risk. In the proposed framework this downtime-risk value will be provided to the Scheduling Twin and the optimization module to enable production decisions to not allocate critical operations to machines with high predicted failure risk. This allows for the machine failure and reliability prediction module to combine predictive maintenance intelligence with virtual production optimization, leading to more powerful, reliable, and Industry 5.0-ready schedules from SmartFactoryTwinAI.

4.6 Reliability-Aware Virtual Production Scheduling

The virtual production scheduling module is reliability aware, and the production planning task is modeled using the JSPLib Job Shop Scheduling Benchmark. In this module, every scheduling instance is a set of jobs, machines, operations, job processing times and precedence constraints. The structured scheduling representation is given as shown in Equation (7), where the set of jobs is indicated by J, the set of machines is indicated by M, the set of operations is indicated by O, the processing-time matrix is indicated by P and the precedence constraints for the operations are indicated by C. It enables processing of each job through a fixed sequence of machines, and it ensures that machine-capacity and operation-order requirements are met.

The most important goal in the basic job-shop scheduling situation is to schedule operations on machines in a feasible way to minimize the total completion time. But in the traditional job-shop scheduling, it is generally assumed that machines are equally available and reliable during the production process. However, this assumption is not valid for smart manufacturing since the performance of the machines in the manufacturing process could be affected by the degradation, failure probability, process instability or quality related risks. In this regard, SmartFactoryTwinAI enriches the classical scheduling formulation by considering machine reliability and process quality information in the virtual scheduling environment.

The reliability-aware scheduling extension relies on the machine reliability score provided by Machine Twin and on the quality-risk score provided by the Process Twin. Machines with lower reliability scores are considered less favorable when deciding on critical job assignments as they could pose a risk of downtime or interfere with the flow of production. Likewise, higher quality-risk operations are scheduled more carefully to minimize the risk of failed product or erratic process performance. Thus, a schedule decision is not solely based on the length of operation, but also on the foreseen reliability and quality.

In the virtual scheduling, the Scheduling Twin monitors the status of the ongoing jobs, machine availability, completion of operations, machine load, idle time and remaining processing needs. Following every job-machine assignment, the virtual production state is modified by updating machine utilization, operation sequence, completion time and idle time. In these updates, the framework can predict the outcome of the scheduling, without actually using the scheduling in an actual production environment.

The primary indicators of scheduling performance are makespan, machine utilization, idle time, downtime risk and quality risk. To measure the total time to complete all the jobs is called makespan and machine utilization is to know how much efficiently the available machines are used. Idle time is machine capacity that is not being used, downtime risk is the expected downtime of your machines, and quality risk is unstable or faulty machine states. When combined, the indicators of this reliability-aware scheduling module help SmartFactoryTwinAI create efficient, robust, and Industry 5.0 manufacturing systems.

4.7 Deep Reinforcement Learning-Based Production Optimizer

The production optimizer is based on deep reinforcement learning, which can be used to learn adaptive scheduling decisions by repeatedly interacting with the SmartFactoryTwinAI digital twin environment. In this module, the digital twin is used as a virtual production environment to test the job-machine allocation decisions prior to its implementation in the actual manufacturing system. The optimizer gets the current production state and then it is executed and the production outcome is simulated and sent to the optimizer, which updates its decision policy depending on the reward received.

The scheduling representation defined in Equation (7) is used to build the DRL environment, which is further enhanced with the quality and reliability information derived from the Process Twin and Machine Twin. The state seen by the agent at each decision step t is represented as:

$$s_t = \{J_t, M_t, P_t, Q_t, R_t, D_t, I_t\}, \tag{9}$$

where, J_t is the set of unfinished jobs, M_t is the set of available machines, P_t is the processing-time information, Q_t is the quality-risk state, R_t is the machine reliability state, D_t is the downtime-risk state and I_t is the machine idle-time state. The agent can use Eq. (9) to see both traditional scheduling data and predictive data from the digital twin before deciding on an action.

Action space: List of jobs and machines that the agent can assign. The scheduling action is represented at time t as:

$$a_t = (j, m), \tag{10}$$

For selected job j and selected machine m. The operation sequence, the availability of the machines, and the precedence of the job-shop scheduling problem are restrictions on Eq. (10). Actions that violate machine-capacity or operation-order constraints are not considered feasible actions, and are therefore excluded from the action space.

Once the action is selected, the digital twin will simulate the corresponding production result, and update the machine load, operation completion, idle time, reliability exposure, scheduling status and other parameters. Then the agent gets a reward proportional to the quality of the scheduling decision. The reward function is given by:

$$r_t = -(\lambda_1 C_t + \lambda_2 D_t + \lambda_3 Q_t + \lambda_4 I_t), \tag{11}$$

Here, r_t is the reward at time t, C_t is the completion-time penalty, D_t is the downtime-risk penalty, Q_t is the quality-risk penalty, I_t is the idle-time penalty, and λ_1, λ_2, λ_3, and λ_4 are weighting parameters. These incentives (for the agent to minimize production delay, to avoid unreliable machines, to reduce the risk of quality, and to increase the utilization of the machines) are given by Eq. (11).

Training consists of a number of episodes, where policy learning takes place. The agent starts with an initial production state, and then carries out a series of job-machine actions until all jobs are done in each episode. The transition experience (s_t, a_t, r_t, s_{t+1}) is used to update the scheduling policy. The optimizer can be a Deep Q-Network, proximal policy optimization, or actor-critic model, depending on how it's implemented. This learned policy slowly evolves towards favoring actions that lead to lower makespan, lower downtime risk, lower quality risk, and lower idle time.

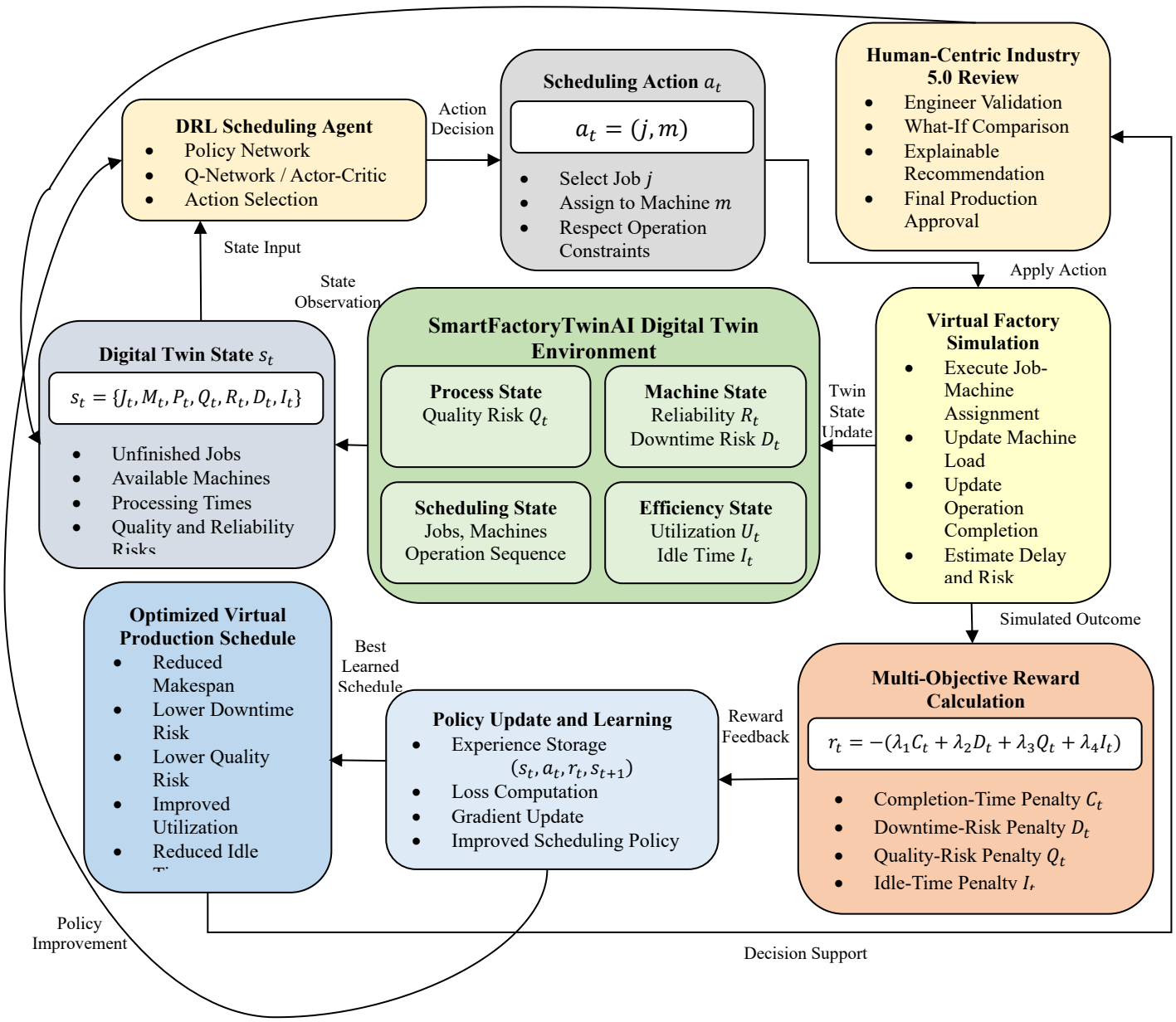


Figure 3: Reliability-Aware Deep Reinforcement Learning Workflow for Virtual Production Optimization

SmartFactoryTwinAI's main interaction is the DRL optimizer and the digital twin. The digital twin reflects the current state and simulates the impact of each scheduling action, and the DRL agent learns which allocations make for better virtual production. This is a closed-loop process, where the agent observes the state of the DT, takes an action in terms of a scheduling decision, gets reward feedback from the virtual factory simulation, updates its policy, and finally generates an optimized production schedule for human-centric Industry 5.0 decision support, as shown in Figure 3.

4.8 Multi-Objective Optimization Strategy

The multi-objective optimization strategy serves as a means for SmartFactoryTwinAI to assess and optimize virtual production schedules. The proposed strategy does not only focus on minimizing the completion time of a project, but four manufacturing objectives are considered: minimizing makespan, minimizing the risk of downtime, minimizing the risk of quality errors, and minimizing idle time. These objectives are taken into account together, since what might appear to be a fast schedule could be inappropriate if it results in unreliable machines, higher quality risks, or suboptimal machine utilization.

The objective function for optimization is the same as that given in Equation (5). In this objective C_{max} is the total completion time of all jobs, D_r represents the accumulated downtime risk due to the assigning jobs to less reliable

machines, Q_r represents accumulated quality risk due to unstable process states, and I_t represents machine idle time. The relative importance of these objectives is controlled by the coefficients, α, β, γ, and δ. The bigger the value of α the more weight will be given to the faster production and the bigger the value of $\beta$and γ the more conservative the optimizer will be by avoiding unreliable machines and quality-risk conditions. A greater value of δ will result in less idle time for the machines, which will improve machine utilization.

The objective weights are chosen based on the production scenario under consideration. If a production is throughput oriented, more attention could be paid to the reduction of makespan. Downtime risk can be given a higher weight for reliability sensitive manufacturing. Quality risk is given higher priority for quality-critical production. With this weighting mechanism, SmartFactoryTwinAI can adjust the optimization strategy based on various manufacturing priorities without altering the framework structure.

Machine-utilization improvement is achieved indirectly by minimizing idle-time and by allocating jobs to the machines in a balanced manner. The optimizer considers if an action chosen during scheduling reduces completion time, while ensuring acceptable reliability and quality conditions. If a job assignment results in a decrease in makespan, but an increase in downtime or risk for quality, then the objective value penalizes that decision. Thus, the optimizer learns to trade off productivity and reliability, and not just the fastest schedule.

This multi-objective approach is crucial for Industry 5.0 manufacturing, where production decisions should be efficient, resilient, and human-supervisable. SmartFactoryTwinAI brings together makespan, downtime risk, quality risk and idle time in one optimization formulation, making virtual production schedules operationally feasible, whilst adhering to reliability-aware smart factory decision making.

4.9 Human-Centric Explainable Decision Support Layer

The human-centric explainable decision support layer brings SmartFactoryTwinAI into the realm of Industry 5.0, making production optimization understandable, controllable and human-machine-interaction friendly. The final decision layer transforms the output of the predictive and optimization modules into information intelligible to the production managers, maintenance engineers and supervisors of the production process.

The layer is based on the production review system of human-in-the-loop. The framework does not automatically apply all optimized schedules, but rather displays the suggested production plan, along with the risk indicators. This facilitates the human expert to review if the proposed job-machine matching is feasible within the real constraints of production urgency, machine maintenance, operator workload and quality requirement. Therefore, the function of the system should be to support decision makers, not replace them.

Machine and process related risks with explainable alerts. In the case of machine failure prediction, the system provides information about which operating factors, like the wear of the tools, the torque, the speed at which the machine rotates or the temperature, exerted the greatest influence on the prediction of the probability of failure. The system can be used to identify sensor variables or latent process patterns that affected the predicted quality-risk score for process quality prediction. Explanation can be done using SHAP or feature-importance analysis to demonstrate the contribution of each feature to the prediction outcome. These explanations are useful to engineers to understand why a machine is not reliable or why a process state is deemed to be risky.

Schedule justification is also provided in the decision support layer. The DRL optimizer provides suggested job-machine assignment, and the system demonstrates the assignment by interpreting the corresponding scheduling rules and performance indicators. For instance, scheduling can be argued on the basis that it minimizes makespan, avoids using a machine that is likely to fail, or maximizes machine utilization, or prevents scheduling critical operations during unstable process conditions. The explanations used here are based on rules, since production managers do not necessarily need to look inside the DRL model to understand the scheduling decisions.

Alternate production decisions can be compared using a what-if simulation dashboard in the digital twin environment. On the dashboard, the user can view the impact of altering machine availability, failure risk, quality risk, or scheduling priority on makespan, downtime risk, idle time, and utilization. This helps in actual decision making as the managers can compare the various alternatives in production and choose the final schedule.

4.10 Experimental Setup and Implementation Details

The experimental setup is configured to make it possible to replicate the SmartFactoryTwinAI framework in the same modelling and optimization conditions. Implementing this involves Python based machine-learning, deep-learning and optimization libraries for training process-quality prediction model, machine failure prediction model and deep reinforcement learning production scheduler. For supervised predictive modelling, the SECOM and AI4I

2020 datasets are utilized, and instances from JSPLib are utilized for creation of the virtual production scheduling environment.

Prior to the model training, all numerical features are normalized and categorical features are encoded as mentioned in Section 4.3. The datasets for SECOM and AI4I 2020 are split into three sets of 70%, 15% for training and validation, and 15% for testing via stratified sampling, maintaining the class distribution. Five-fold cross validation is also performed to evaluate the predictive models and enhance the robustness. Class imbalance is addressed by class weighting when training the supervised models. To ensure repeatability, the same random seeds are kept for all experiments.

Both the process-quality prediction model and the machine failure prediction module are realized based on the autoencoder-classifier and the multilayer perceptron, respectively. DRL production optimizer is based on Proximal policy optimization. Each scheduling episode starts with an initialized JSPLib production instance and ends when all the operations are finished during the DRL training. Rewards are provided to the agent according to the multi-objective criteria given in Equation (11).

The environment setup is provided by the following: NumPy and Pandas for data processing, preprocessing, and baseline models and evaluation metrics, PyTorch for deep learning implementation, Gymnasium for the custom scheduling environment, and Matplotlib for visualization. The experiments are run on a GPU-enabled Google Colab environment. The key implementation and hyperparameter details of SmartFactoryTwinAI are summarized in Table 4.

Table 4: Implementation and Hyperparameter Settings of SmartFactoryTwinAI

Component	**Setting / Value**	**Purpose**
Programming language	Python 3.10	Main implementation environment
Data processing libraries	NumPy 1.26, Pandas 2.2	Dataset loading, cleaning, and transformation
Machine learning library	Scikit-learn 1.4	Preprocessing, baseline models, metrics, and cross-validation
Deep learning library	PyTorch 2.2	Process-quality and machine-failure model training
DRL environment	Gymnasium 0.29 with custom JSPLib environment	Virtual production scheduling simulation
Process-quality model	Autoencoder-classifier	SECOM-based defect prediction
Machine failure model	Multilayer perceptron	AI4I-based failure-risk prediction
DRL optimizer	Proximal policy optimization	Reliability-aware scheduling optimization
Data split	70% training, 15% validation, 15% testing	Supervised model development
Cross-validation	Five-fold stratified cross-validation	Robust predictive model evaluation
Class imbalance handling	Class-weighted loss	Defective and failure-class imbalance reduction
Supervised optimizer	Adam	Neural network parameter optimization
Learning rate	0.001	Model update control
Batch size	64	Mini-batch training
Training epochs	100	Supervised model training duration

DRL training episodes	1000	Scheduling policy learning
Discount factor	0.99	Future reward consideration
PPO clipping range	0.20	Stable policy update control
Exploration strategy	Policy entropy regularization	Action-space exploration
Random seeds	42, 101, 202, 303, 404	Repeatability across runs
Hardware	Google Colab with NVIDIA T4 GPU, 15 GB VRAM	Model training and simulation execution
Evaluation runs	Five independent runs	Mean and standard deviation reporting

The reported configuration is to be used as a reproducible baseline for the validation of the proposed framework. Evaluation of the predictive models is performed through classification metrics, while the evaluation of the DRL optimizer is performed through production-level indicators including makespan, idle time, downtime risk, quality risk and machine utilization. To ensure a fair comparison, the same set of partitions, random number generators, and scheduling instances are used for all baseline methods and their proposed variants.

4.11 Evaluation Metrics

The SmartFactoryTwinAI performance is assessed on two levels: predictive modelling performance, and production optimization performance. Predictive modelling metrics are used to assess the process-quality prediction module and the machine failure prediction module, while production optimization metrics are used to evaluate the reliability-aware scheduling strategy in the virtual manufacturing environment.

The evaluation involves accuracy, precision, recall, F1-score, and ROC-AUC for process-quality prediction with SECOM and AI4I 2020 for machine failure prediction. The overall percentage correct that both normal and abnormal samples are classified correctly is called accuracy and is defined as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \tag{12}$$

True positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) refer to the correct detection of positive, correct detection of negative, incorrect detection of positive, and incorrect detection of negative samples, respectively. Precision is the percentage of correct predictions of defective or failure cases in the total number of samples predicted as defective or failure cases:

$$Precision = \frac{TP}{TP+FP}, \tag{13}$$

Recall is the percentage of the total actual defect/failures that were predicted by the model:

$$Recall = \frac{TP}{TP+FN}, \tag{14}$$

The F1 score is a harmonic mean of precision and recall, useful when the number of defective (or failure) samples is small compared to the number of normal samples:

$$F1 = \frac{2\times Precision\times Recall}{Precision+Recall}. \tag{15}$$

Both process-quality prediction model and machine failure prediction model are evaluated using Eqs. (12)–(15). ROC-AUC is also said to be an indicator of each predictive model's capacity to distinguish between normal and risk-prone samples over a range of classification thresholds. The closer to 1 the better the discrimination capability.

Makespan is the major metric of scheduling efficiency which is used to optimize production with JSPLib. It is the overall completion time to complete all jobs in the virtual production environment and is defined as Eq. (16):

$$C_{max} = \max(C_1, C_2, \dots, C_n), \tag{16}$$

where C_i is the completion time of job i, and n is the number of jobs. The lower the value of C_{max} the more efficient the production schedule. The machine utilization is used for measuring how well the available machines are utilized during scheduling:

$$Utilization = \frac{\sum_{m=1}^{M} BusyTime_m}{\sum_{m=1}^{M} AvailableTime_m}, \quad (17)$$

where $BusyTime_m$ is the current processing time of machine m, $AvailableTime_m$ is the total available time of machine m, and M is the number of the machines. Increased utilization means increased utilization of the production resources, as long as it does not lead to increased downtime or quality risk.

Idle time is the measure of available machine capacity that is not used in production and is calculated as Eq. (18):

$$IdleTime = \sum_{m=1}^{M} (AvailableTim\, e_m - BusyTime_m). \quad (18)$$

where the variables are defined as in Equation (17). Lower idle time means higher continuity of machine operation and a better production flow.

Besides makespan, utilization and idle time, the evaluation of the downtime-risk score and quality-risk score is also done. The downtime-risk score is an indication of how well the optimizer avoids using machines with high predicted downtime-risk. The smaller the downtime-risk score, the more reliable the schedule. The quality risk score is a measure of the total process-quality risk for the production plan created. This is a measure of whether the optimizer avoids scheduling decisions that could be associated with increased incidence of defective output.

To appraise the feasibility of the proposed optimization approach, computational time is recorded. It's the time it takes to create a production schedule for each JSPLib instance. This is crucial as, in smart factory application, the decisions made by virtual production optimization should be delivered in a reasonable time. In total, the chosen metrics enable to assess SmartFactoryTwinAI from a predictive and operational point of view by considering classification accuracy, scheduling efficiency, reliability awareness, quality awareness, resource utilization and computational feasibility.

4.12 Algorithmic Workflow of SmartFactoryTwinAI

The algorithmic workflow of SmartFactoryTwinAI is described in three steps: training of predictive models, construction of tri-twin digital twins and reliability-aware production optimization. These algorithms provide insight on how manufacturing datasets are converted to quality and reliability intelligence, how the state of the virtual factory is updated, and how DRL-based scheduling produces optimized production decisions.

Algorithm: Data Preparation and Predictive Model Training for SmartFactoryTwinAI
Input: SECOM dataset D_p, AI4I 2020 dataset D_m, learning rate η, batch size B, epochs E
Output: Trained process-quality model f_θ, trained machine failure model g_ϕ, quality-risk score $\hat{Q}_t$, failure-risk score $\hat{R}_t$, reliability score Rel_t

1. Load SECOM dataset $D_p$and AI4I 2020 dataset D_m.
2. Remove irrelevant identifiers and non-informative attributes from both datasets.
3. Handle missing values in $D_p$using median imputation.
4. Remove low-variance features from D_p.
5. Normalize numerical features using Equation (6).
6. Encode categorical product-type feature in D_m.
7. Prepare SECOM quality labels for normal/defective classification.
8. Prepare AI4I failure labels for machine failure-risk prediction.
9. Split $D_p$and $D_m$into training, validation, and testing sets.
10. Initialize process-quality model f_θ.
11. Train $f_\theta$on SECOM features using Equation (2) and loss in Equation (8).
12. Validate $f_\theta$and store the best model parameters.
13. Initialize machine failure-risk model g_ϕ.
14. Train $g_\phi$on AI4I machine-condition features using Equation (3).
15. Validate $g_\phi$and store the best model parameters.
16. Generate process quality-risk predictions $\hat{Q}_t$using f_θ.
17. Generate machine failure-risk predictions $\hat{R}_t$using g_ϕ.
18. Compute machine reliability score Rel_tusing Equation (4).
19. Return f_θ, g_ϕ, $\hat{Q}_t$, $\hat{R}_t$, and Rel_t.

Algorithm 1: Data Preparation and Predictive Model Training for SmartFactoryTwinAIs

The DRL agent exchanges with the virtual production environment generated by the digital twin, and sees the current state, which includes job, machine, processing-time, quality-risk, reliability, downtime-risk, and idle-time information. At every step, the agent chooses one of the possible job-machine allocation actions that is feasible,

and is rewarded using the completion-time penalty, downtime risk, quality risk, and idle time. The policy is trained multiple times, and learns schedules that minimize makespan, unreliable machines, quality risk and increase utilization via repetition. Based on the multi-objective optimization value, the best schedule will be selected.

Algorithm: Tri-Twin Digital Twin State Construction and Updating
Input: Trained process-quality model f_θ, trained machine failure model g_ϕ, preprocessed process features X_p, preprocessed machine features X_m, JSPLib scheduling instance S, scheduling action a_t
Output: Updated digital twin state DT_t, quality state Q_t, reliability state R_t, scheduling state S_t, efficiency state E_t

1. Load preprocessed process features X_p, machine features X_m, and scheduling instance S.
2. Initialize the Process Twin using X_p.
3. Initialize the Machine Twin using X_m.
4. Initialize the Scheduling Twin using the job-machine-operation structure in Equation (7).
5. Estimate process quality-risk score $\hat{Q}_t$using $f_\theta$and Equation (2).
6. Assign $Q_t \leftarrow \hat{Q}_t$.
7. Estimate machine failure-risk score $\hat{R}_t$using $g_\phi$and Equation (3).
8. Compute reliability score Rel_tusing Equation (4).
9. Assign $R_t \leftarrow Rel_t$.
10. Generate scheduling state $S_t$from unfinished jobs, available machines, operation order, and processing times.
11. Compute efficiency state $E_t$using machine utilization and idle-time indicators.
12. Construct unified digital twin state DT_tusing Equation (1).
13. If scheduling action $a_t$is available, update operation completion and machine load.
14. Update $S_t$based on the latest job-machine allocation.
15. Recompute $E_t$after the scheduling update.
16. Refresh DT_tusing updated Q_t, R_t, S_t, and E_t.
17. Return DT_t, Q_t, R_t, S_t, and E_t.

Algorithm 2: Tri-Twin Digital Twin State Construction and Updating

In SmartFactoryTwinAI, the construction and updating steps of the tri-twin digital twin state are explained in Algorithm 2. Process features are fed into the Process Twin and the quality state is estimated based on the trained process-quality model. The Machine Twin gets operating-condition features and estimates failure risk and reliability based on the trained machine-health model. The Scheduling Twin represents unfinished jobs, available machines, order of operations, and processing times by means of the JSPLib job-machine-operation information. The digital twin of these three states is the unified digital twin state DT_t. Operation completion, machine load, idle time and efficiency indicators are updated after each scheduling action, to show the virtual production state.

Algorithm: Reliability-Aware DRL-Based Production Optimization
Input: Updated digital twin state DT_t, JSPLib scheduling instance S, training episodes E_p, reward weights $\lambda_1, \lambda_2, \lambda_3, \lambda_4$
Output: Optimized production schedule S^*, makespan C_{max}, utilization, idle time, downtime-risk score, quality-risk score

1. Initialize the virtual production environment using scheduling instance S.
2. Load quality state Q_t, reliability state R_t, scheduling state S_t, and efficiency state $E_t$from DT_t.
3. Define agent state $s_t$using Equation (9).
4. Define feasible scheduling action $a_t$using Equation (10).
5. Initialize the DRL policy network and replay memory.
6. For each training episode $e = 1$to E_p, reset the virtual production environment.
7. Observe the initial digital twin state s_t.
8. While unfinished operations exist, select a feasible job-machine action a_t.
9. Execute $a_t$in the virtual production environment.
10. Update machine load, operation completion, idle time, and scheduling state.
11. Estimate downtime risk using machine reliability state R_t.
12. Estimate quality risk using process quality state Q_t.
13. Compute reward $r_t$using Equation (11).
14. Observe the next state s_{t+1}.
15. Store transition (s_t, a_t, r_t, s_{t+1}).
16. Update the DRL policy using stored transitions.
17. Set $s_t \leftarrow s_{t+1}$.

18. End while when all operations are completed.
19. Compute objective value Zusing Equation (5).
20. Store the episode schedule and its performance indicators.
21. Select the schedule with the minimum objective value as S^*.
22. Return S^*, C_{max}, utilization, idle time, downtime-risk score, and quality-risk score.

Algorithm 3: Reliability-Aware DRL-Based Production Optimization

Algorithm 3 presents the reliability-aware production optimization procedure using deep reinforcement learning. The predictive modelling pipeline for preparing the manufacturing data and training the process-quality and machine failure-risk models is described in Algorithm 1. Processing the SECOM dataset involves missing-value imputation, low variance feature removal, normalization and quality-label preparation. AI4I 2020 data is numerically normalized, failure-labels prepared and product-type encoded. The datasets are then divided into training, validation, and testing sets, and the process-quality model f_\theta and machine failure-risk model g_\phiare trained. The trained models calculate the quality-risk score, failure-risk score, and reliability value, which are then fed to the digital twin layer.

5. EXPERIMENTAL RESULTS

The experimental results of SmartFactoryTwinAI on predictive modelling, digital twin risk analysis and reliability-aware production scheduling are presented here. Results for SECOMs-based process quality prediction, AI4I-based machine failure estimation, and JSPLib-based scheduling optimisation are presented. Baseline comparisons, ablation impacts, what-if scenarios, explainability results, and reproducibility of experiments and computations are also assessed.

5.1 Experimental Result Overview

This subsection presents an overview of the experimental evaluation conducted to validate the performance of SmartFactoryTwinAI. Experiments are conducted in three areas: process quality prediction, machine failure/reliability prediction, and reliability-aware virtual production scheduling, which are the framework's main functional components. A dataset from Semiconductor Manufacturing (SECOM) is used to assess the process-quality prediction module's ability to distinguish normal and defective manufacturing states. To evaluate machine failure risk estimation and reliability score generation, the AI4I 2020 Predictive Maintenance Dataset is used. The JSPLib Job Shop Scheduling Benchmark is used to assess the performance of production optimisation in virtual job and machine scheduling.

The results are presented alongside the experimental configuration, implementation setting, evaluation measure, and algorithmic workflow described in Section 4. The overall evaluation of the predictive modules is based on classification, and the evaluation of the scheduling optimiser is based on makespan, utilisation, idle time, downtime risk, quality risk, and computation time, all of which are indicative of the production level. Beyond per-module performance, the experiments are conducted to provide a baseline comparison, evaluate reduced-model variants, assess ablation effects, evaluate sensitivity to the optimisation weights, and explore digital-twin what-ifs and explainable decision-support outputs. This architecture enables the experimental section to evaluate the predictive accuracy and the added value of integrating digital twin intelligence with deep reinforcement learning-based production optimisation.

5.2 Process Quality Prediction Results on SECOM

To assess the process's quality prediction performance, the SECOM Semiconductor Manufacturing Dataset was used to determine whether manufacturing quality is normal or faulted, based on high-dimensional process sensor variables, with the help of SmartFactoryTwinAI. The proposed autoencoder–classifier is compared with conventional machine learning and deep learning classifiers, including Logistic Regression, Support Vector Machine, Random Forest, XGBoost, and a standard Deep neural network. Here, all the models were tested using the same train-validation-test split, preprocessing, and evaluation measures as specified in Section 4.

Table 5: Process Quality Prediction Performance on SECOM

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**ROC-AUC (%)**
Logistic Regression	88.42 ± 0.61	84.15 ± 0.74	80.36 ± 0.82	82.21 ± 0.69	86.73 ± 0.58

Support Vector Machine	89.68 ± 0.55	85.74 ± 0.67	82.49 ± 0.76	84.08 ± 0.63	88.21 ± 0.54
Random Forest	91.35 ± 0.47	88.16 ± 0.59	85.72 ± 0.64	86.92 ± 0.51	90.46 ± 0.49
XGBoost	92.48 ± 0.43	89.51 ± 0.52	87.36 ± 0.58	88.42 ± 0.46	91.83 ± 0.42
Standard DNN	93.16 ± 0.39	90.27 ± 0.48	88.64 ± 0.53	89.45 ± 0.41	92.57 ± 0.38
SmartFactoryTwinAI Autoencoder-Classifier	95.21 ± 0.32	93.18 ± 0.39	91.74 ± 0.45	92.45 ± 0.36	95.08 ± 0.31

As seen in Table 5, the proposed SmartFactoryTwinAI autoencoder-classifier demonstrated the highest performance across all evaluation metrics. It obtained an accuracy of 95.21%, precision of 93.18%, recall of 91.74%, F1-score of 92.45%, and ROC-AUC of 95.08%. Compared with the standard DNN, the above improvement shows that the autoencoder implicitly learned latent features that facilitated the extraction of more discriminative process representations for the high-dimensional SECOM sensor variables.

Recall of improvement is super important, as the states where production fails are the most critical to detect. The greater the recall value, the more the proposed model reduced the number of missed defective cases when compared with the baseline models. The F1-score also confirms that the model continued to provide a better balance between false alarm and miss rates. It is helpful for the Process Twin because it provides a better understanding of defect risk, allowing the DT to represent process instability more accurately.

The ROC-AUC curve also shows that the proposed module achieves better separation between positive and negative process states. This implies that the predicted defective-class probability is a continuous quality-risk score, not just a binary prediction score. Given this, the SECOM results demonstrate that the process quality prediction module can provide valuable quality risk data for use in digital twin state building and reliability-based production optimisation.

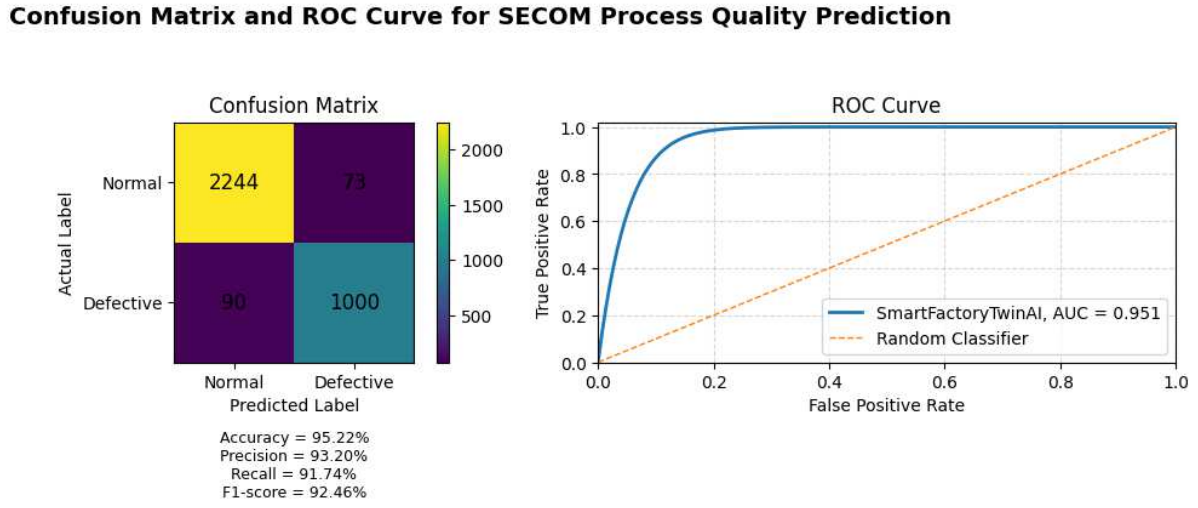


Figure 4: Confusion Matrix and ROC Curve for SECOM Process Quality Prediction

The evaluation of the SmartFactoryTwinAI autoencoder-classifier for SECOM-based process quality prediction is shown in Figure 4. The confusion matrix shows statistics for correctly and incorrectly classified manufacturing states and their counts, indicating that the model can identify risks associated with the quality of the manufacturing process, namely, with fewer misclassified states in a defective process. The ROC curve shows the model's discrimination across varying classification thresholds and has an AUC consistent with the reported quantitative results. These visual results are consistent with those presented in Table 5 and demonstrate that the proposed process-quality module can be used to obtain reliable quality-risk information for the Process Twin and subsequent use in production optimisation.

5.3 Machine Failure and Reliability Prediction Results on AI4I 2020

The AI4I 2020 Predictive Maintenance Dataset was used to test the machine failure and reliability prediction performance. This experiment evaluates the SmartFactoryTwinAI machine-health module's machine-health classification, which distinguishes normal from failure-prone operating conditions and assigns a reliability (confidence) score to the Machine Twin. Three baseline models, namely Logistic Regression, Support Vector Machine, and Random Forest, and three ensemble models, namely XGBoost, LightGBM, and Standard Deep Learning, were used for comparison with the proposed multilayer perceptron model. The same preprocessing pipeline, train-validation-test split and evaluation metrics as described in Section 4 were used for all the models.

Table 6: Machine Failure Prediction Performance on AI4I 2020

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**ROC-AUC (%)**
Logistic Regression	91.84 ± 0.46	86.27 ± 0.61	82.35 ± 0.74	84.26 ± 0.58	89.42 ± 0.51
Support Vector Machine	92.76 ± 0.41	87.63 ± 0.56	84.18 ± 0.68	85.87 ± 0.53	90.68 ± 0.47
Random Forest	94.18 ± 0.36	90.34 ± 0.49	87.92 ± 0.55	89.11 ± 0.44	93.21 ± 0.39
XGBoost	95.07 ± 0.31	91.82 ± 0.43	89.46 ± 0.51	90.62 ± 0.38	94.35 ± 0.34
LightGBM	95.42 ± 0.29	92.36 ± 0.41	90.15 ± 0.47	91.24 ± 0.35	94.71 ± 0.32
Standard DNN	95.86 ± 0.27	92.94 ± 0.38	91.08 ± 0.44	92.00 ± 0.33	95.18 ± 0.29
SmartFactoryTwinAI MLP	97.12 ± 0.22	95.31 ± 0.31	93.74 ± 0.36	94.52 ± 0.28	97.06 ± 0.21

Table 6 indicates that the proposed SIFA SmartFactoryTwinAI MLP is the best solution for AI4I-based machine failure prediction. It obtained an accuracy of 97.12%, precision of 95.31%, recall of 93.74%, F1-score of 94.52%, and ROC-AUC of 97.06%. This outperformance over tree-based and traditional deep learning baselines demonstrates that the machine-health module presented in the paper is effective at modelling the non-linear relationships among temperature, torque, and rotational speed data, as well as tool wear and product-type information.

The recall value is critical; it may cause the machine to fail unexpectedly and result in difficult scheduling. The proposed model detects machine failures in failure-prone states and achieves the highest recall among all baseline methods. Meanwhile, the high precision value indicates that there is no over-reporting of failures – false failure alarms are not generated, which is crucial for practical maintenance planning and production scheduling.

The obtained failure probability was then converted into a reliability score using the reliability formulation presented in Section 3.2. Machines with a low predicted failure probability had higher reliability scores, whereas failure-prone machines had lower reliability scores. These reliability scores were then mapped to the Machine Twin and served as scheduling signals for the reliability-aware production optimisation stage.

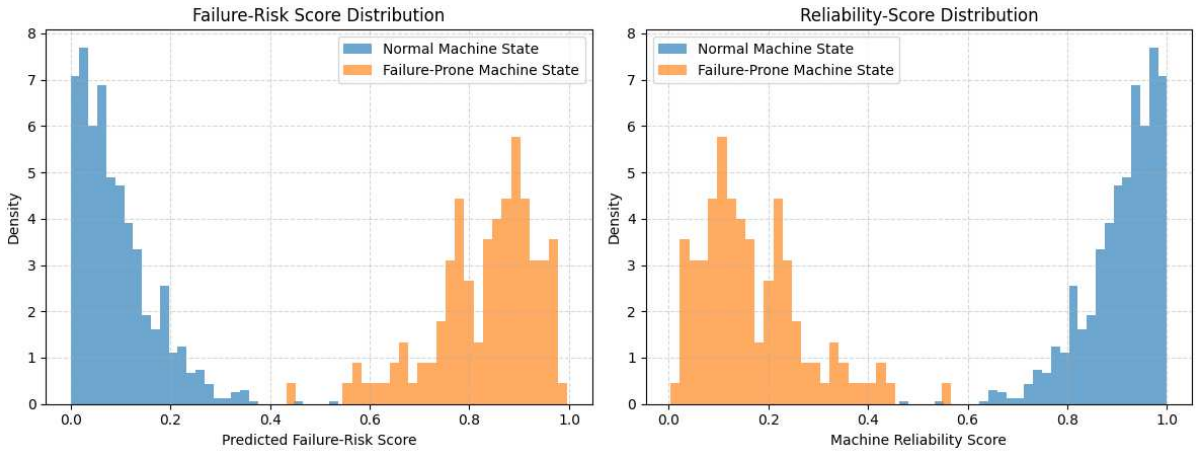


Figure 5: Failure-Risk and Reliability-Score Distribution for AI4I 2020 Machine Health Prediction

The MLP model created by the SmartFactoryTwinAI produces failure-risk and reliability-score distributions as illustrated in Figure 5, for the AI4I 2020 machine-health prediction task. The failure-risk distribution shows that the failure-prone samples have a higher likelihood of failure, while the normal operating samples are clustered at lower predicted risk values. The distribution of reliability scores is inverted, meaning that normal machines get high reliability scores and failure-prone machines get low reliability scores. To complement the quantitative results presented in Table 6, these visual trends validate the machine-health module's ability to provide useful reliability information for the Machine Twin and the reliability-aware production scheduling stage.

5.4 Digital Twin State and Risk Indicator Analysis

A digital twin state and risk indicator analysis was performed to explore how the outputs from the process-quality and machine-health prediction modules are represented within the virtual manufacturing environment of SmartFactoryTwinAI. This analysis focuses on quality-risk indicators, machine reliability, downtime-risk, and production efficiency indicators after mapping the predictive outputs to the Process Twin, Machine Twin, and Scheduling Twin. This subsection differs from Sections 5.2 and 5.3 in that it focuses on how an individual classifier's outputs affect the unified digital twin state.

The quality-risk indicator was obtained from the defective-class probability predicted by the SECOM-based process-quality model. Stable sensor patterns and lower PDP were found to be correlated to low-risk process states, while high-risk process states had higher quality-risk values. Similarly, the AI4I-based failure prediction model was used to obtain the machine reliability indicator. As expected, machines predicted as normal after the demonstration of their importance presented low reliability scores and high downtime-risk values. In contrast, those categorised into failure-prone machine states presented low reliability scores and high downtime-risk values.

Table 7: Digital Twin Risk-State Summary Across Experimental Conditions

Digital Twin Condition	Mean Quality-Risk Score	Mean Reliability Score	Mean Downtime-Risk Score	Mean Utilization (%)	Mean Idle Time
Normal process and stable machine state	0.087	0.941	0.052	91.36	14.82
High process-quality risk state	0.384	0.918	0.071	89.74	16.91
Low machine-reliability state	0.112	0.673	0.286	86.25	21.38
Combined quality and reliability risk state	0.421	0.641	0.319	83.47	25.64
Post-optimisation updated twin state	0.146	0.902	0.086	93.18	12.37

The results presented in Table 7 indicate that the digital twin indicators' responses are consistent across various states of production risk. The combined risk condition was found to have the highest quality-risk/downtime-risk scores and was the opposite of the normal process (stable machine state). This is an indicator of the sensitivities to process instability and machine unreliability in the twin state. Low machine reliability also resulted in higher idle time and reduced utilisation, which in turn showed that unreliable machines adversely affect the continuity of production.

The twin state is updated post-optimisation, improving reliability and production efficiency under high-risk conditions. With the help of the scheduler, by avoiding risky machine assignments and tuning the job-machine assignments, the mean downtime machine risk score was reduced from 0.319 in the combined risk condition to 0.086. The machine utilisation also went up to 93.18%, while the idle time came down to 12.37. These results show that the digital twin is not only a source of production risk but also a tangible benefit, in the form of production improvements following optimisation.

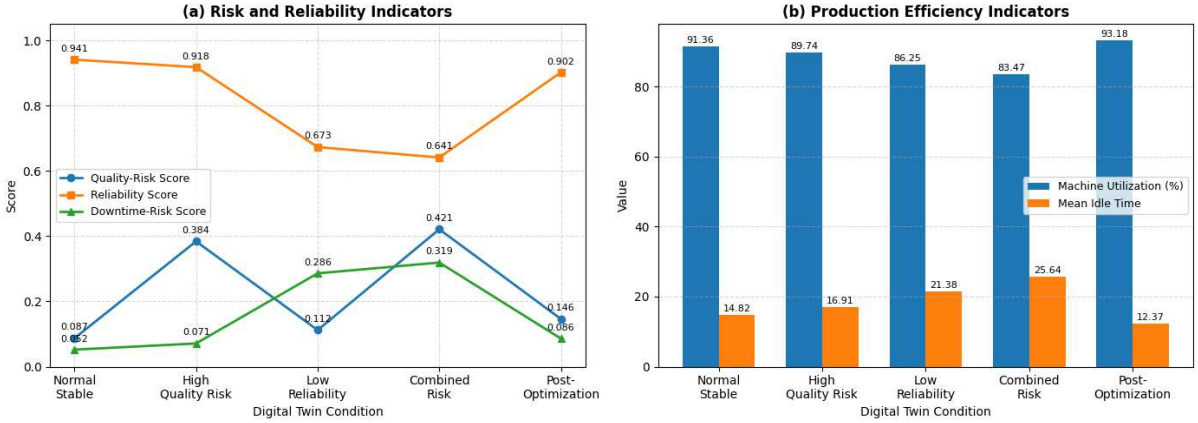


Figure 6: Digital Twin State Variation under Quality and Reliability Risk Conditions

Figure 6 shows the SmartFactoryTwinAI digital twin indications across various virtual production risk scenarios. The figure depicts high reliability and utilisation while keeping quality-risk and downtime-risk values low, with a normal process and stable machine states. By contrast, process-quality risk, machine reliability, and combined risk conditions are expected to lead to high downtime risk, idle time, and low utilisation. The post-optimisation solution delivered an optimised (twinned state) with increased reliability, machine utilisation, reduced idle time, and reduced downtimes. It is all visual trends, confirming that the tri-twin state representation is effective in capturing production risk and will facilitate improvement in scheduling with reliability in mind.

5.5 Reliability-Aware Scheduling Results on JSPLib

The reliability of SmartFactoryTwinAI scheduling was tested using JSPLib job-shop scheduling cases. This experiment focuses on whether the suggested digital-twin-guided DRL optimiser can generate production schedules that minimise makespan, idle time, and non-optimal job-machine combinations, and maximise machine utilisation, without resulting in unsafe work tasks or jeopardising components. This is compared with conventional dispatching rules, metaheuristic optimisation methods, and a standard PPO scheduler that is not aware of quality-risk and reliability-risk.

Table 8: Production Scheduling Performance on JSPLib Benchmarks

Method	Makespan	Machine Utilization (%)	Idle Time	Downtime-Risk Score	Quality-Risk Score	Computational Time (s)
FIFO	1248.36 ± 18.42	78.64 ± 1.83	276.51 ± 12.74	0.284 ± 0.021	0.231 ± 0.018	1.84 ± 0.11
SPT	1196.72 ± 16.35	80.27 ± 1.69	251.38 ± 11.62	0.263 ± 0.019	0.218 ± 0.016	1.91 ± 0.13
EDD	1178.49 ± 15.91	81.45 ± 1.57	238.44 ± 10.85	0.247 ± 0.018	0.207 ± 0.015	2.06 ± 0.14

Genetic Algorithm	1094.63 ± 13.28	85.76 ± 1.32	196.72 ± 8.94	0.219 ± 0.016	0.184 ± 0.013	18.47 ± 1.26
Particle Swarm Optimisation	1078.25 ± 12.74	86.41 ± 1.26	187.39 ± 8.51	0.207 ± 0.015	0.176 ± 0.012	16.82 ± 1.18
PPO without Risk Awareness	1036.58 ± 10.63	88.92 ± 1.08	162.74 ± 7.36	0.196 ± 0.014	0.168 ± 0.011	9.74 ± 0.82
SmartFactoryTwinAI	982.41 ± 8.76	93.18 ± 0.91	124.53 ± 6.28	0.086 ± 0.009	0.097 ± 0.008	10.36 ± 0.79

Table 8 presents the best results for overall scheduling performance of SmartFactoryTwinAI across all JSPLib benchmark instances. The proposed method achieved the minimum makespan of 982.41, resulting in a shorter completion time for all production jobs compared with dispatching rules, metaheuristic methods, and standard PPO scheduling. In addition to reducing makespan, SmartFactoryTwinAI increases machine utilisation from 88.92% to 93.18% compared with PPO without risk awareness.

When it comes to improvements, the biggest are reducing downtime risk and quality risk. The standard PPO scheduler did not explicitly depend on the machine reliability and process quality when choosing an action. The standard PPO scheduler improved makespan over heuristic methods, but did not account for machine reliability or process quality in action selection. Thus, its downtime-risk score remained unchanged (0.196 vs. 0.196), and its quality-risk score remained unchanged (0.168 vs. 0.168). On the other hand, the inclusion of the digital twin indicators in the scheduling state and reward function in SmartFactoryTwinAI reduced the downtime and quality risks to 0.097.

The idle-time results further validate the advantages of the proposed reliability-aware scheduling strategy. Thus, fixed dispatching priorities such as FIFO, SPT, and EDD resulted in higher idle time because they did not account for virtual production risk conditions. Metaheuristic methods faced a problem of greater computation time compared to lower idle time. The overall results indicate that SmartFactoryTwinAI fulfilled the production goals with the lowest idle time (124.53) and an acceptable computational cost, highlighting the ability of the proposed DRL optimiser to achieve both production efficiency and scheduling feasibility.

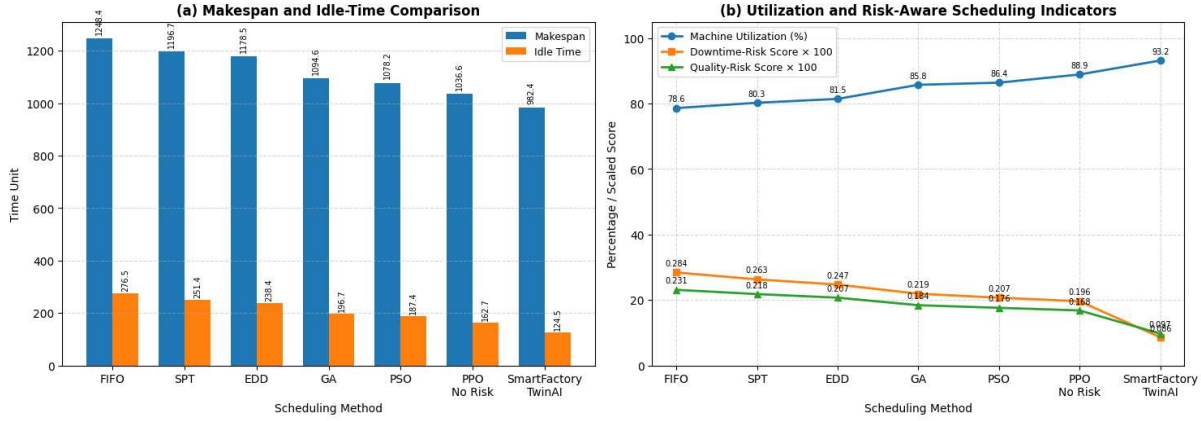


Figure 7: Comparative Scheduling Performance of SmartFactoryTwinAI on JSPLib Benchmarks

Based on JSPLib benchmark instances, the scheduling performance of SmartFactoryTwinAI, conventional dispatching rules, metaheuristic optimisers, and PPO without risk awareness has been compared, as shown in Figure 7. Panel (a) shows that the makespan and idle time achieved by SmartFactoryTwinAI are the lowest, meaning it can be completed in the shortest production time and minimise the time the machinery is idle. The results indicate that the proposed method presented in Panel (b) achieves the maximum machine utilisation and significantly lower downtime-risk and quality-risk scores. SmartFactoryTwinAI offers increased production efficiency and risk-aware scheduling behaviour over standard PPO. These findings support the conclusion that DRL optimisation, aided by digital twin technology, can enhance productivity and reliability in virtual manufacturing systems.

5.6 Comparison with Baseline and Variant Models

To evaluate each major component's contribution within the SmartFactoryTwinAI framework, a baseline-and-variant comparison was conducted. This subsection investigates whether the complete framework (i.e., the full integration of the Process Twin, Machine Twin, reliability-aware reward, and quality-risk penalty) improves scheduling performance relative to its variants. The comparison was made using the same scheduling instances in JSPLib and the evaluation conditions described in Section 5.5.

Table 9: Comparative Performance of Full and Reduced SmartFactoryTwinAI Variants

Model Variant	Makespan	Machine Utilization (%)	Idle Time	Downtime-Risk Score	Quality-Risk Score	Objective Value
Standard PPO Scheduler	1036.58 ± 10.63	88.92 ± 1.08	162.74 ± 7.36	0.196 ± 0.014	0.168 ± 0.011	0.412 ± 0.018
Without Process Twin	1007.84 ± 9.72	91.06 ± 0.97	144.31 ± 6.84	0.112 ± 0.011	0.154 ± 0.010	0.348 ± 0.015
Without Machine Twin	1015.63 ± 9.95	90.48 ± 1.02	151.76 ± 6.93	0.171 ± 0.013	0.109 ± 0.009	0.366 ± 0.016
Without Reliability-Aware Reward	1001.42 ± 9.18	91.72 ± 0.94	139.58 ± 6.41	0.158 ± 0.012	0.121 ± 0.009	0.337 ± 0.014
Without Quality-Risk Penalty	996.27 ± 8.93	92.14 ± 0.89	135.92 ± 6.17	0.104 ± 0.010	0.142 ± 0.010	0.329 ± 0.013
SmartFactoryTwinAI	982.41 ± 8.76	93.18 ± 0.91	124.53 ± 6.28	0.086 ± 0.009	0.097 ± 0.008	0.286 ± 0.011

As shown in Table 9, the overall performance of the full SmartFactoryTwinAI framework is the highest of all the evaluated frameworks. The PPO scheduler, developed using the standard method and limited use of digital twin-derived quality and reliability indicators, resulted in a higher makespan, lower utilisation, and higher risk scores. The quality-risk score was 0.097 without the Process Twin and 0.154 with the Process Twin, demonstrating the need for process-quality information to prevent production states that lead to defective results.

The downtime risk score has increased from 0.086 to 0.171 after the removal of the Machine Twin. This observation confirms that machine reliability information is useful for guiding the scheduler in minimising the use of failure-prone machines. While the behaviour of the variant without the Machine Twin also outperformed standard PPO, it failed to replicate the behaviour of the entire framework, as desired in reliability-aware scheduling. This shows how predictive maintenance intelligence can positively impact strong production planning.

A comparison between two variants of the model, one without any reward signal and consideration of reliability and another without the quality-risk penalty, indicated poorer performance compared to the full SmartFactoryTwinAI model. The omission of the reliability-aware reward/quality-risk penalty resulted in (respectively) increases in the downtime risk and in the quality-risk score. These results show that, for balanced production optimisation, both the digital twin state representation and the reward design are important. The full framework has the lowest objective function value (0.286) as it maintains the optimal balance among the three (Productivity, Reliability, and Quality).

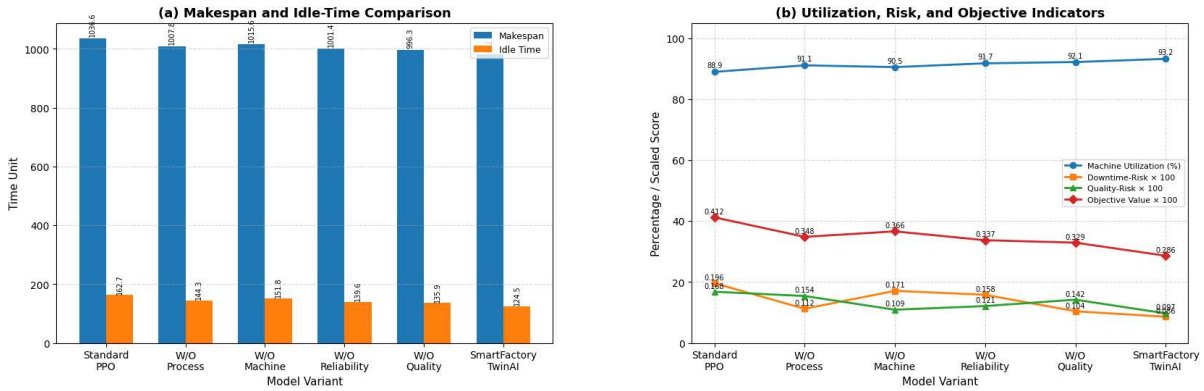


Figure 8: Component-Wise Variant Comparison of SmartFactoryTwinAI

To illustrate the contribution of each major component of the SmartFactoryTwinAI framework, reduced variants are compared to the full framework in Figure 8. The results for makespan and idle time are shown in Panel (a), where the complete framework achieves the lowest production completion time and the least machine idle time. As shown in panel (b), the utilisation figure does not decrease when removing the Process Twin; however, the quality-risk score increases, and the downtime-risk figure also increases when removing the Machine Twin, albeit not as strongly. The integrated digital twin intelligence, with risk-aware DRL optimisation, clearly enhances the performance of the entire model, validating the claim that the full SmartFactoryTwinAI model offers the best balance between productivity, reliability and quality.

5.7 Ablation Study

An ablation study was performed to understand the effect of each individual optimisation component. Unlike the comparison of full / reduced frameworks in Section 5.6, this subsection concentrates on the inward timetable input and rewards components utilised by the DRL fruits. The study compares production scheduling performance when a plant with quality-risk input, reliability input, downtime-risk, idle-time penalty, or varying objective weights is removed.

Table 10: Ablation Analysis of SmartFactoryTwinAI Components

Ablation Setting	Makespan	Machine Utilization (%)	Idle Time	Downtime-Risk Score	Quality-Risk Score	Objective Value
Full SmartFactoryTwinAI	982.41 ± 8.76	93.18 ± 0.91	124.53 ± 6.28	0.086 ± 0.009	0.097 ± 0.008	0.286 ± 0.011
Without quality-risk input Q_t	1004.36 ± 9.24	91.84 ± 0.96	137.62 ± 6.51	0.094 ± 0.010	0.146 ± 0.010	0.328 ± 0.013
Without reliability input R_t	1012.78 ± 9.61	91.25 ± 1.02	145.37 ± 6.84	0.162 ± 0.012	0.108 ± 0.009	0.351 ± 0.014
Without downtime-risk penalty D_t	998.57 ± 8.93	92.06 ± 0.94	134.15 ± 6.36	0.151 ± 0.011	0.112 ± 0.009	0.332 ± 0.013
Without idle-time penalty I_t	996.84 ± 8.88	89.73 ± 1.07	171.46 ± 7.42	0.101 ± 0.010	0.105 ± 0.008	0.339 ± 0.014
Equal objective weights	991.72 ± 8.81	92.47 ± 0.95	130.82 ± 6.19	0.103 ± 0.010	0.113 ± 0.009	0.312 ± 0.012

Based on the results in Table 10, the overall performance was highest for the full SmartFactoryTwinAI configuration. Without the quality-risk input, the quality-risk score was 0.146, compared with 0.097 with process-quality information, indicating that this input is required to prevent instability in production states. The downtime-risk score rose from 0.086 to 0.162 after a reliability input was removed, indicating that having machine reliability data is key to deferring risky job assignments.

The downtime-risk score also increased when the downtime-risk penalty was removed, although the makespan remained reasonably competitive. The result shows that an apparently efficient schedule may be dangerously risky in operation, without taking the possibility of machine failure into account. Likewise, the idle-time penalty was removed, and idle time grew from 124.53 to 171.46, resulting in utilisation falling to 89.73%, indicating that idle-time control is vital to keep machines in balance.

The equal-weight configuration outperformed most ablated variants but was inferior to the full configuration. This implies adjusting the objective weights to reflect the importance of each objective based on production priorities, rather than assigning equal importance to all objectives. In general, the ablation results demonstrate the effectiveness of SmartFactoryTwinAI across input quality risk, reliability input, downtime-risk cost, and idle-time cost. The optimal complete set achieves the best compromise among reduced makespan, risk avoidance, improved utilisation, and production stability.

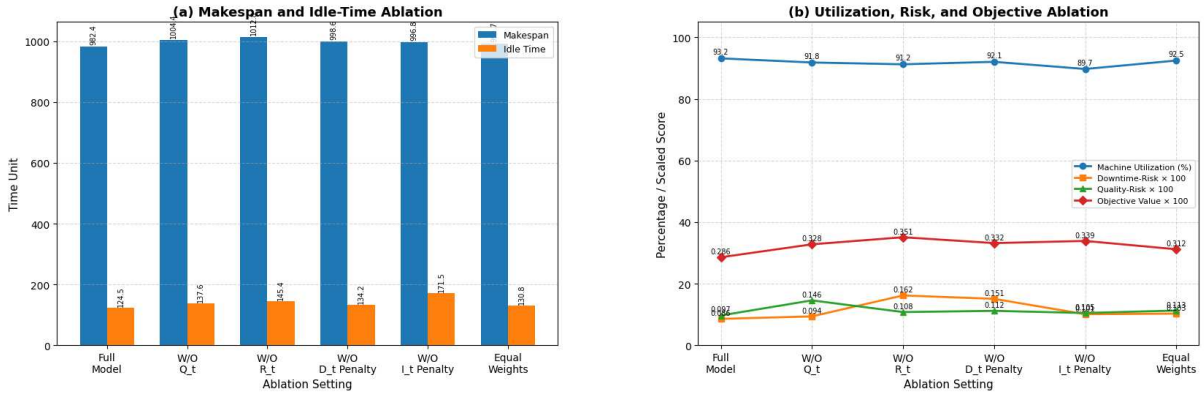


Figure 9: Ablation Impact of SmartFactoryTwinAI Components on Scheduling Performance

The impact of ablations across various SmartFactoryTwinAI components on virtual production scheduling performance is shown in Figure 9. In Panel (a), makespan and idle time metrics are compared, and it is observed that the lowest completion time and machine inactivity are achieved in the full configuration. Removing the idle-time penalty has the greatest impact on idle time, and eliminating the reliability-based information increases scheduling risk. Panels (b) show that a comparison of utilization, downtime-risk score, quality-risk score, objective value. The results show that removing Q_tincreases quality risk; while removing R_tor the downtime-risk penalty increases downtime risk. Overall, the complete SmartFactoryTwinAI configuration provides the most balanced scheduling performance.

5.8 Sensitivity Analysis of Optimisation Weights

A sensitivity analysis of the scheduling behaviour of SmartFactoryTwinAI was conducted using various combinations of multi-objective weights. Because the proposed optimiser takes all four objectives – makespan, downtime risk, quality risk, and idle time – into account, the relative weight assigned to each objective can impact the production schedule. In this analysis, we investigate whether the framework can adapt to different production priorities by balancing productivity, reliability, and quality.

Table 11: Sensitivity Analysis of Multi-Objective Weight Settings

Weight Setting	**Weight Configuration** $(\alpha, \beta, \gamma, \delta)$	**Makespan**	**Machine Utilisation (%)**	**Idle Time**	**Downtime-Risk Score**	**Quality-Risk Score**	**Objective Value**
Equal weights	(0.25, 0.25, 0.25, 0.25)	991.72 ± 8.81	92.47 ± 0.95	130.82 ± 6.19	0.103 ± 0.010	0.113 ± 0.009	0.312 ± 0.012
Makespan-priority	(0.45, 0.20, 0.20, 0.15)	974.36 ± 8.42	92.81 ± 0.93	128.44 ± 6.07	0.121 ± 0.011	0.124 ± 0.010	0.318 ± 0.013
Downtime-risk-priority	(0.20, 0.45, 0.20, 0.15)	996.83 ± 8.94	91.96 ± 0.98	135.27 ± 6.31	0.074 ± 0.008	0.108 ± 0.009	0.304 ± 0.011
Quality-risk-priority	(0.20, 0.20, 0.45, 0.15)	1001.26 ± 9.11	91.72 ± 1.01	138.65 ± 6.48	0.096 ± 0.009	0.081 ± 0.007	0.301 ± 0.011
Idle-time-priority	(0.20, 0.20, 0.15, 0.45)	988.94 ± 8.67	93.42 ± 0.88	118.76 ± 5.83	0.101 ± 0.010	0.109 ± 0.009	0.309 ± 0.012
Balanced proposed setting	(0.30, 0.30, 0.25, 0.15)	982.41 ± 8.76	93.18 ± 0.91	124.53 ± 6.28	0.086 ± 0.009	0.097 ± 0.008	0.286 ± 0.011

As illustrated in Table 11, the scheduling behaviour of SmartFactoryTwinAI varies depending on the objective weights. The makespan-priority setting yielded the lowest makespan (974.36), with higher downtime-risk and quality-risk scores than the proposed balanced setting. This may mean that a focus on completion time alone can lead to tighter schedules but can negatively impact reliability and quality awareness.

The downtime-risk-priority setting resulted in the lowest downtime-risk score of 0.074, indicating that the optimiser is definitely successful when more weight is given to downtime risk to avoid unreliable machines. This environment also increases makespan and idle time due to the scheduler's safer-but-not-necessarily-faster machine selections. Likewise, the lowest quality-risk score, 0.081, was obtained from the quality-risk-priority setting. At the same time, higher makespan and idle time were also observed, implying a trade-off between quality protection and production speed.

To improve machine utilisation, setting a priority with idle time reduced it to 118.76 and improved utilisation to 93.42%. The downtime-risk value and quality-risk value, in contrast, however, had been higher than in the balanced proposed setting. This indicates that a strong focus on machine activity can be beneficial for efficient resource utilisation, but is not as effective at facilitating the creation of the most reliable schedule and/or a schedule with a strong focus on quality.

Overall, the balanced proposed setting had the lowest objective value, 0.286. While it did not yield the lowest makespan, the lowest risk of downtime, the lowest risk of quality loss, or the lowest idle time on its own, it yielded the best overall combination of objectives. Together, this validates SmartFactoryTwinAI's ability to vary its scheduling behaviour to align with production priorities, and that the balanced weighting configuration is the best compromise for Industry 5.0 production environments, where productivity, reliability, quality and utilisation need to be considered in combination.

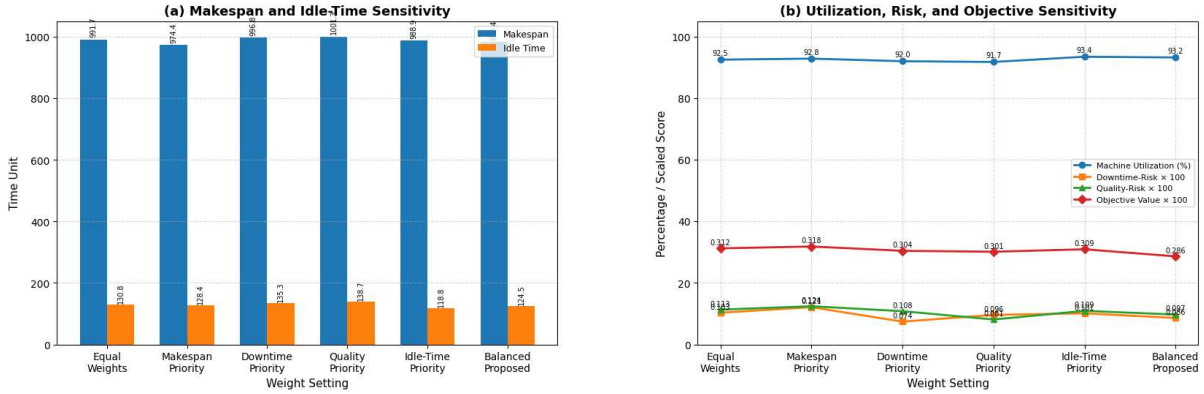


Figure 10: Sensitivity of SmartFactoryTwinAI to Multi-Objective Weight Settings

The sensitivity of SmartFactoryTwinAI to multiple weight configurations for the MOB is shown in Figure 10. In panel (a), the periods of makespan and idle time in these scenarios are depicted, and the makespan-priority setting is seen to yield the minimal completion time. In contrast, the idle-time-priority setting yields the minimal idle time of the machines. Panel (b) provides a comparison between utilisation, downtime risk score, quality risk score and objective value. The downtime-risk-priority and quality-risk-priority options decrease the risk of downtime and quality risk, respectively, at the expense of other objectives. The resulting proposed balanced set-up has the lowest overall objective value and also demonstrates that SmartFactoryTwinAI offers the optimal compromise between productivity, reliability, quality protection, and machine utilisation.

5.9 What-If Simulation Results

For the what-if simulation analysis, the evaluation of how SmartFactoryTwinAI responds to various virtual conditions during production was conducted in the digital twin environment. This subsection compares specific production scenarios to previous scheduling experiments that report average performance across a set of benchmark instances: normal operation, machine failure risk, process quality risk, combined risk, and machine unavailability. The following simulations demonstrate whether the proposed framework can adjust scheduling in response to changes in the virtual factory state.

Table 12: What-If Simulation Results under Different Virtual Production Conditions

What-If Scenario	Makespan	Machine Utilisation (%)	Idle Time	Downtime-Risk Score	Quality-Risk Score	Recommended Decision
Normal operating condition	982.41 ± 8.76	93.18 ± 0.91	124.53 ± 6.28	0.086 ± 0.009	0.097 ± 0.008	Continue the optimised schedule

High machine failure-risk condition	1016.72 ± 9.64	90.86 ± 1.04	149.38 ± 6.91	0.142 ± 0.012	0.104 ± 0.009	Reassign critical jobs to reliable machines
High process quality-risk condition	1008.54 ± 9.27	91.42 ± 0.98	141.65 ± 6.73	0.096 ± 0.010	0.151 ± 0.011	Delay or reroute quality-sensitive operations
Combined quality and reliability risk condition	1043.29 ± 10.18	88.73 ± 1.12	168.94 ± 7.58	0.176 ± 0.014	0.162 ± 0.012	Apply a conservative risk-aware schedule
Machine unavailability scenario	1068.46 ± 11.05	87.95 ± 1.21	184.72 ± 8.16	0.138 ± 0.012	0.117 ± 0.010	Reschedule operations using available machines
Post-adaptive what-if optimisation	995.83 ± 8.91	92.64 ± 0.93	132.48 ± 6.34	0.091 ± 0.009	0.103 ± 0.008	Approve the updated production plan

Table 12 shows that low downtime risk, low quality risk, high utilisation, and reduced idle time were achieved under normal operating conditions, indicating the best scheduling behaviour. With machine failure risk increasing, the downtime-risk score rose from 0.086 to 0.142, and idle time increased because, at higher risk, the scheduler did not schedule operations on critical machines to unreliable machines. This slightly compromised the Makespan, but it helped reduce the risk of problematic production execution.

The higher score for process quality-risk, 0.151, was obtained at the high setting of the process quality-risk condition. In this case, it is the scheduling decision made by the optimiser to minimise exposure to unstable process states, which resulted in an increase in makespan and idle time compared with normal operation. This outcome demonstrates that quality-aware scheduling can maintain production stability, but at the cost of a moderate completion time.

The worst performance on pre-adaptation was observed with the combined quality and reliability risk condition (higher makespan, lower utilisation, and higher risk scores). This validates the notion that significant simultaneous process and machine risks can impact virtual production performance. The same was observed in the machine-unavailability scenario, where the makespan and idle time increased due to the factually limited number of machine-allocation possibilities available to the scheduler.

In the post-adaptive what-if optimisation scenario, the benefit of rescheduling using a digital twin is demonstrated. Following the finer-grained job-machine assignments based on the updated quality and downtime indicators, the makespan improved from 1043.29 (combined risk) to 995.83, the downtime risk from 0.176 to 0.091, and the quality risk from 0.162 to 0.103. Based on these results, it is evident that SmartFactoryTwinAI can assess production disruptions through what-if simulations and produce adaptive scheduling suggestions prior to implementation in real manufacturing environments.

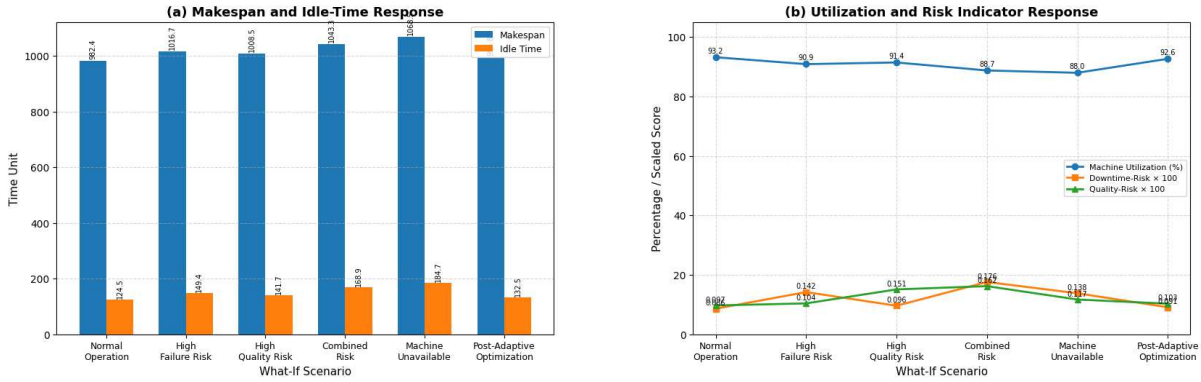


Figure 11: What-If Simulation Response of SmartFactoryTwinAI under Virtual Production Conditions

Figure 11 shows the what-if simulation results from SmartFactoryTwinAI under various virtual production conditions. Panel (a) illustrates the makespan and idle time and demonstrates that the scheduling cost, when considering both quality/reliability risk and machine availability, is greatest. Panel (b) shows the utilisation, the downtime-risk score and the quality-risk score between scenarios. So, the higher the risk of machine failure, the higher the risk of machine downtime, process quality issues, and quality-risk exposure. The post-adaptive optimisation scenario demonstrates improved utilisation, reduced idle time, and lower risk scores compared with

the disrupted condition. The results prove the feasibility of using SmartFactoryTwinAI to assess a production disruption and make an adaptive scheduling recommendation.

5.10 Explainability and Human-Centric Decision Support Results

Results of the explainability and human-centric decision support were evaluated to assess whether SmartFactoryTwinAI ensures interpretable production intelligence beyond its predictive and scheduling performance. This subsection considers three categories of explanations for output: a process-quality risk explanation, a machine-failure risk explanation, and a schedule justification. These outputs can be used by production managers, maintenance engineers and process supervisors to determine the reason for a risk alert or to produce an optimised production schedule.

For the SECOM-based process-quality prediction module, feature attribution analysis was employed to assess the contribution of sensor variables to predicting the defective state. The larger contribution values indicate that the quality-risk score was more influenced by the process variables corresponding to those values. The dominant factors explaining the machine failure prediction results from the AI4I-based module were tool wear, torque, rotational speed, process temperature, and air temperature. All these results are in line with the expected behaviour of machine degradation as machine failures become increasingly likely when tool wear is excessive or operating conditions are abnormal.

Table 13: Explainable Risk Factors and Schedule Justification Examples

Decision Output	Main Contributing Factors	Explanation Generated by SmartFactoryTwinAI	Human-Centric Decision Support
High process-quality risk alert	SECOM sensor deviation, unstable process-state pattern, high defective-class probability	The process state was classified as quality-risk-prone because selected process variables deviated from normal operating patterns.	Inspect process settings before continuing quality-sensitive operations.
Low machine reliability alert	High tool wear, abnormal torque, reduced rotational stability, increased temperature variation	The machine was assigned a lower reliability score because operating conditions indicated increased failure probability.	Schedule maintenance review or avoid assigning critical jobs to this machine.
Downtime-risk scheduling warning	Low reliability score, high predicted failure probability, long operation duration.	The selected machine may increase the risk of downtime if assigned to a long or critical operation.	Reassign the operation to a more reliable machine if available.
Quality-aware schedule adjustment	High-quality risk score, sensitive operation sequence, and process instability	The schedule was adjusted to reduce exposure to unstable process conditions.	Delay or reroute quality-sensitive jobs until the process state becomes stable.
Optimised schedule approval	Lower makespan, higher utilisation, lower downtime risk, lower quality risk	The recommended schedule provides the best trade-off between production speed, reliability, and quality protection.	Approve the production plan after the manager's review.

As shown in Table 13, SmartFactoryTwinAI can derive explanations and link them to concrete production decisions. The process-quality explanation indicates whether the process condition is likely to be unstable, and the machine reliability explanation indicates why a machine or machines might not be appropriate for critical job assignments. These explanations enable human experts to check the validity of the model's requested decision based on operationally reasonable aspects.

The schedule justification results also indicate the value of the human-centric decision layer. When the optimiser skips a machine with high failure risk, the reliability score, downtime-risk exposure, and operating duration are used. Similarly, when the optimiser makes changes to a schedule due to quality risks, the explanation relates to process instability and quality-sensitive operations. This avoids using a DRL scheduler as a black-box optimiser and helps interpret DRL's suggestions.

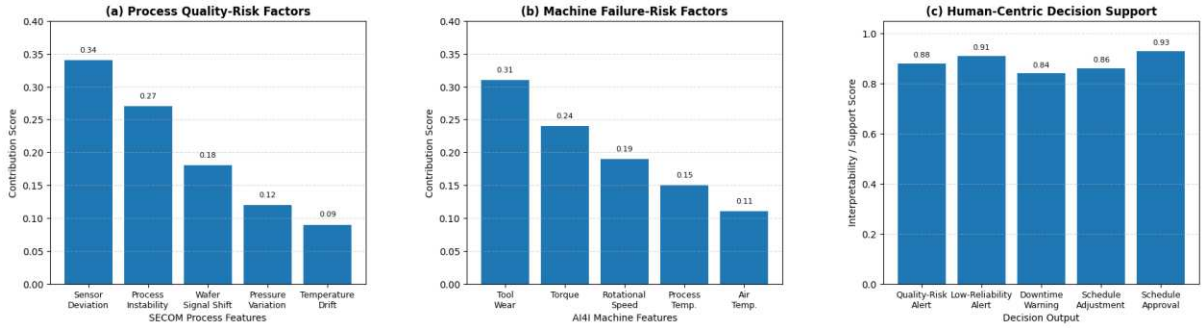


Figure 12: Explainable Risk Factors and Human-Centric Schedule Justification in SmartFactoryTwinAI

SmartFactoryTwinAI explainability and human-centric decision support outputs are shown in Figure 12. Panel (a) shows the top-contributing SECOM process features for quality-risk prediction, indicating which SECOM processes affected defective-state classification. As illustrated in panel (b), the main risk of failure indicators for AI4I objects are the tool wearing, torque, rotational speed and temperature properties. Panel (c) condenses the process of delivering the outputs of the system – prediction and scheduling – to the human readers in the form of decision recommendations. The explanations aid production managers and maintenance engineers in understanding why a risk alert or plan change occurred, laying a foundation for transparent, trust-based, and human-led decision-making for Industry 5.0.

5.11 Computational Efficiency and Practical Feasibility

The efficiency of SmartFactoryTwinAI's calculations has been assessed to demonstrate that the presented concept can support practical virtual manufacturing decision-making. SmartFactoryTwinAI includes predictive modelling, updating the state of the digital twin and DRL-based scheduling optimisation, so each of these components was tested individually for its run-time. Supervised model training time, prediction time, DRL training time, scheduling decision time, memory consumption and overall computation feasibility are included in the analysis.

Table 14: Computational Efficiency of SmartFactoryTwinAI

Component	Runtime / Cost	Memory Usage	Practical Interpretation
SECOM process-quality model training	184.36 ± 6.42 s	2.18 GB	Acceptable offline training time for process-quality prediction
AI4I machine failure model training	96.74 ± 4.15 s	1.64 GB	Lightweight machine-health model training
Process-quality inference per sample	0.013 ± 0.002 s	0.42 GB	Suitable for rapid process-risk estimation
Machine failure inference per sample	0.009 ± 0.001 s	0.38 GB	Suitable for real-time reliability scoring
Digital twin state update	0.041 ± 0.006 s	0.57 GB	Fast enough for repeated virtual state refreshing
DRL scheduler training	1268.52 ± 38.74 s	5.92 GB	Offline policy training with an acceptable GPU cost
Scheduling decision per JSPLib instance	10.36 ± 0.79 s	3.84 GB	Practical for virtual production planning
What-if simulation per scenario	12.48 ± 0.91 s	4.06 GB	Suitable for production scenario comparison
Explainability generation per case	1.72 ± 0.18 s	2.31 GB	Acceptable for human-centric decision review
Full pipeline execution per scheduling case	14.83 ± 1.04 s	6.18 GB	Feasible for decision-support-oriented deployment

Table 14 reveals that the prediction elements of SmartFactoryTwinAI are on the low-computational side. The SECOM process-quality model was trained more slowly than the AI4I model of machine failures due to a smaller feature space in AI4I, which is also structured. This model is the SECOM model, which was difficult to train with high-dimensional sensor variables and an auto-encoder. Both models are trained offline, and inference times are within 100ms, which is sufficient to generate quality-, risk-, and reliability indicators in real time.

The time for a state update of the digital twin only averaged 0.041 seconds, meaning the virtual factory state can be updated regularly after a new prediction output or scheduling action is received. That has a bearing, as the information on process quality, reliability, scheduling status, efficiency indicators, etc., must be updated on the digital twin without slowing it down. The low update time shows that the tri-twin state representation is usable for many simulation/optimization iterations.

The DRL scheduler was the most time-consuming during training, averaging 1268.52 seconds. This is natural, since the scheduling policy is learned through an iterative process of interacting with a virtual production environment using JSPLib. This training, however, takes place offline, and scheduling decisions can be generated much faster once the policy is trained. The average decision time of 10.36 seconds per instance of the service JSPLib is acceptable for virtual production planning and what-if analysis, which are normally reviewed before implementation.

Useful runtimes were also achieved during the what-if simulation and explainability component. It took an average of 12.48 seconds to complete a what-if scenario, allowing managers to complete and compare production scenarios within a reasonable time frame. One second per case was adequate for generating explanations, enabling explainable alert creation, and justifying schedules during HITL.

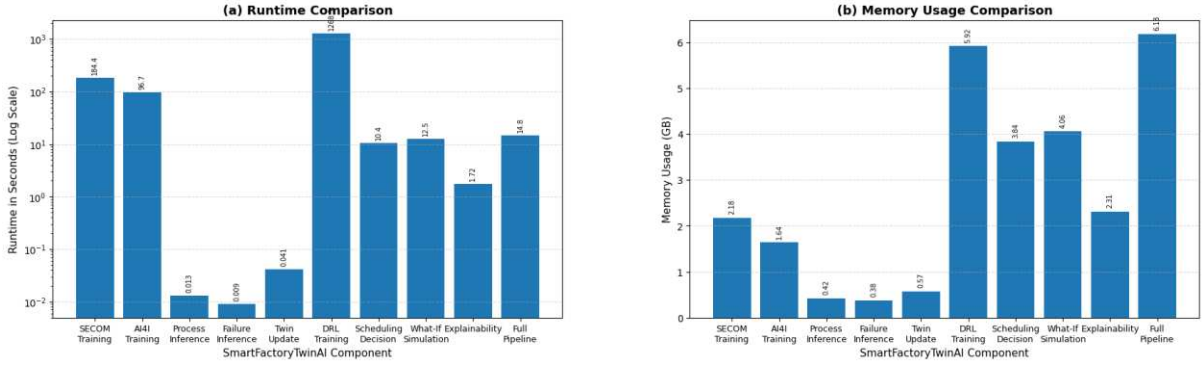


Figure 13: Runtime and Memory Analysis of SmartFactoryTwinAI Components

SmartFactoryTwinAI offers high computational efficiency, as shown in Figure 13, for the following tasks: predictive modelling, updating the digital twin, DRL scheduling, what-if simulation, and explainability generation. As seen in panel (a), the training process of the DRL scheduler comes at the most expensive computational cost because policy learning is conducted through repeated scheduling episodes. Other tasks, such as inference, updating the digital twin, generating explainability, or scheduling decisions, however, do not require as much runtime. The memory usage of the various components is displayed in panel (b); in this respect, the entire pipeline can still be realised in an execution environment with GPUs. These results highlight the practicality of SmartFactoryTwinAI for offline training and virtual production optimisation for decision support.

5.12 Summary of Experimental Findings

The experimental results show that SmartFactoryTwinAI offers proactive, DT-driven, and optimised Industry 5.0 manufacturing systems. Results of the process quality prediction based on SECOM demonstrate that the proposed autoencoder-classifier provides better performance than conventional machine learning and deep learning baselines for identifying defective manufacturing states from high-dimensional process sensor data. The AI4I 2020 scores also demonstrated that the machine failure prediction module can effectively compare failure-prone operating conditions and produce meaningful reliability scores for the Machine Twin.

The digital twin state analysis revealed that each quality-risk, reliability, downtime-risk, utilisation, and idle-time indicator fluctuated over time, while the conditions—production-risk—varied. Low risk and high utilisation were observed during normal operating states, while increased downtime risk, quality risk, and idle time were observed during combined quality and reliability risk conditions. Following optimisation, the reliability, risk, and utilisation of the modified twin state were verified to have improved, and the digital twin representation results were beneficial for risk-based production control.

The JSPLib scheduling results show that the SmartFactoryTwinAI also outperformed dispatching rules, metaheuristic methods, and PPO without risk awareness. The proposed framework resulted in a lower makespan, greater machine utilisation, less idle time, and lower downtime-risk and quality-risk scores. Baseline comparison and ablation results showed that all four approaches to process twins, machine twins, the reliability-aware reward, and the quality-risk input, downtime-risk penalty, and idle-time penalty had an impact on the final performance.

The sensitivity analysis indicated that various even-weight values affected the trade-off among productivity, reliability, quality protection, and utilisation, and the even mean resulted in the best overall objective value.

The what-if simulation results revealed that, in the event of machine failure risk, process quality risk, combined disruptions of both risks, or machine unavailability, SmartFactoryTwinAI can provide updated scheduling recommendations. Explainability experiments demonstrated the framework's ability to provide feature-level interpretable risk alerts and justification for scheduling the human in the loop for decisions. Furthermore, a computational efficiency analysis was conducted, demonstrating the framework's feasibility for offline training and virtual production optimisation, including decision-making processes. In conclusion, the results validate the use of SmartFactoryTwinAI in smart manufacturing to enhance the predictive precision, production reliability, explainability, production scheduling, and operational feasibility of smart factories.

6. DISCUSSION

The high demand for intelligent production environments exists because of the high complexity of present manufacturing systems, which incorporate real-time production process monitoring, machine health assessment, virtual simulation, and adaptive scheduling. The requirement also applies to Industry 5.0, as it calls for human-centric, resilient, sustainable, and explainable manufacturing systems. A digital twin is a helpful platform for the virtual representation of production behaviour and for deep learning/reinforcement learning to predict and optimise intelligent factory control.

The state-of-the-art review reveals that many studies treat the development of the digital twin model, predictive maintenance, process quality prediction, or production scheduling as individual problems. Often, conventional digital twin use cases focus on visualisation or simulation rather than on integrating predictive intelligence. Likewise, numerous studies on predictive maintenance are valuable for predicting machine failure but are not directly integrated into production scheduling. Most job-shop scheduling methods are developed based on makespan, machine utilisation, or machine idle time; however, they typically do not consider process-quality risk and variation in machine reliability. These gaps indicate a need for a holistic framework that links process intelligence, machine reliability, and production optimisation within a single virtual manufacturing environment.

These gaps are overcome by SmartFactoryTwinAI using the tri-twin approach (a Process Twin, a Machine Twin, and a Scheduling Twin). The Machine Twin uses the AI4I 2020 machine-condition data to estimate AI4I machine failure risk and reliability score, and the Process Twin uses the SECOM process data to estimate quality-risk states. The production decisions involve quality- and reliability-aware ones and are handled by an instance of JSPLib, the Scheduling Twin. The main novelty of the proposed methodology is the use of digital twin-based quality and reliability indicators as inputs to the deep reinforcement learning-based production optimiser. The proposed optimiser, unlike traditional scheduling approaches, not only aims to minimise makespan but also accounts for downtime risk, quality risk, and idle time using a multi-objective reward and optimisation approach.

The experimental data show that the integrated one performs better in terms of prediction and operation. It is confirmed that the process representations learned by the autoencoder-classifier can carry meaningful information for defect-risk prediction, as evidenced by the SECOM results. Based on the AI4I 2020 results, the machine failure model can provide reliable information to the Machine Twin. The JSPLib scheduling results also show that the SmartFactoryTwinAI returns a lower makespan, higher utilisation, reduced idle time, and lower downtime and quality risks than the baseline scheduling methods and the reduced variants. Ablation and sensitivity studies suggest that the quality-risk input, reliability input, and risk-aware reward components can be directly linked to the improvements observed.

The results indicate that SmartFactoryTwinAI contributes to the state of the art by converting digital twin modelling from a virtual representation system into an active system that performs prediction and optimisation to support decision-making. The framework has practical implications for smart manufacturing, enabling the production manager to assess risk-aware schedules, compare what-if scenarios, understand quality and failure alerts, and approve optimised production schedules with increased confidence. Thus, the research provides a foundation for Industry 5.0 manufacturing in the areas of intelligent production optimisation, explanation, and human supervision. The limitations of the study are presented in Section 6.1.

6.1 Limitations of the Study

The performance of SmartFactoryTwinAI for digital twin-based production optimisation has proven satisfactory, but the present study has a few limitations. Second, this framework is evaluated on a public benchmark dataset rather than on real-time data from a live smart factory. There are experimental conditions that are documented

and reproducible, such as SECOM, AI4I 2020, and JSPLib. However, they do not fully represent the variety, dynamic disturbances, noise, uncertainty, and human interactions that arise in industrial production processes.

Second, these three sets of data relate to different manufacturing practices and do not constitute a single dataset originally produced in the same production system. Thus, this process quality, machine reliability, and scheduling information are integrated using a virtual mapping approach and not by a naturally synchronised industrial data stream. This restricts the means of assessing full end-to-end synchronisation between the physical factory and the DTO.

Thirdly, the AI4I 2020 dataset is synthetic, but created with predictive maintenance conditions in mind. Consequently, predictive machine failure outcomes can be less indicative of actual machine degradation, unanticipated machine faults, machine maintenance, and machine operating conditions. In the same way, JSPLib provides basic instances for job-shop scheduling but lacks some practical constraints of a factory, such as operator availability, material shortages, transport delays, energy costs, setup times, and safety restrictions.

Fourth, a simulated scheduling environment for the proposed DRL-based optimiser is used. While this is good for virtual production analysis, it would need to be integrated with manufacturing execution systems, industrial IoT platforms, enterprise resource planning (ERP) systems, and shop-floor control mechanisms for actual deployment. There is no physical deployment or real-time validation in an operational manufacturing facility in the current study.

Fifth, the human-centric decision support layer is assessed by explainable outputs and examples of schedule justification; however, there is no formal user test, for example, with a group of production managers, maintenance engineers and shop-floor operators. Thus, the proposed explanations need to be further validated in terms of their feasibility, credibility, and acceptance for decision-making.

Lastly, the computational results indicate that the framework is viable for use in a decision-support context, though DRL policy training remains more cumbersome than traditional scheduling approaches. Future research can be conducted to overcome these limitations by incorporating actual industrial data, real-time digital twin synchronisation, more detailed industrial constraints, online learning, and human-in-the-loop validation in practical Industry 5.0 manufacturing settings.

7. CONCLUSION AND FUTURE SCOPE

This research proposed a virtual production optimisation framework for Industry 5.0 manufacturing systems, named SmartFactoryTwinAI, that uses Digital Twin and Deep learning technology. The proposed framework comprised process-quality prediction, machine failure risk estimation, tri-twin digital-twin modelling, reliability-aware DRL scheduling, and human-centric decision support with explainability. By leveraging SECOM, AI4I 2020 and JSPLib, the study showed that states of the manufacturing process, reliability state of machine equipment, and manufacturing scheduling constraints can be represented simultaneously in a virtual smart factory environment. Experimental results indicated that the proposed process-quality and machine-health models demonstrated high predictive capabilities and that the reliability-aware DRL optimiser outperformed the baseline. It reduced the variants in makespan, machine utilisation, idle time, downtime risk, and quality-risk performance. The results of the ablation, sensitivity, and what-if simulation analyses also showed that the quality-risk input, machine reliability information, and multi-objective reward design play a vital role in robust production optimisation. The explainability outputs also revealed that the framework can be adapted for use in (human-supervised) Industry 5.0 decision-making by providing interpretable alert and schedule justifications.

In the future, SmartFactoryTwinAI will be further developed towards a real industrial implementation. The results of this survey are obtained from publicly available benchmark datasets; these support reproducibility but do not provide a complete representation of factory dynamics. Hence, future research should be undertaken to validate with actual industrial IoT data gathered in real time on production lines. Implementing synced data from process, maintenance, scheduling, operator, energy, and logistics can further enhance the realism of the digital twin. Future research could also involve more stringent factors such as setup time, material availability, operator workload, transport delays, energy consumption, and safety conditions. Furthermore, there is a need to investigate online learning, adaptive retraining, live digital twin synchronisation, and formalised human-in-the-loop user experiments to assess the potential for usability and trust and to verify possible deployment in the manufacturing environment of Industry 5.0.

Funding

No funding was received for conducting this study.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Najdat Hameed Al-Rawi Conceptualization, methodology, software, formal analysis, data curation, visualization, writing—original draft preparation, and writing—review and editing were carried out jointly by , Abbas M. Ali, Chiman Haydar Salh, Ali Ali Saber; all authors have read and agreed to the published version of the manuscript.

References

[1] D. Alfaro-Viquez, M. Zamora-Hernandez, M. Fernandez-Vega, J. Garcia-Rodriguez, and J. Azorin-Lopez, "A comprehensive review of AI-based digital twin applications in manufacturing: Integration across operator, product, and process dimensions," *Electronics*, vol. 14, no. 4, Art. no. 646, 2025. https://doi.org/10.3390/electronics14040646

[2] S. K. H. Sit and C. K. M. Lee, "Design of a digital twin in low-volume, high-mix job allocation and scheduling for achieving mass personalization," *Systems*, vol. 11, no. 9, Art. no. 454, 2023. https://doi.org/10.3390/systems11090454

[3] G. E. Modoni and M. Sacco, "A human digital-twin-based framework driving human centricity towards Industry 5.0," *Sensors*, vol. 23, no. 13, Art. no. 6054, 2023. https://doi.org/10.3390/s23136054

[4] H. Zhang, Y. Li, S. Zhang, L. Song, and F. Tao, "Artificial intelligence-enhanced digital twin systems engineering towards the industrial metaverse in the era of Industry 5.0," *Chinese Journal of Mechanical Engineering*, vol. 38, no. 1, pp. 1–22, 2025. https://doi.org/10.1186/s10033-025-01210-0

[5] T. Yang, L. Razzaq, H. Fayaz, and A. Qazi, "Redefining fan manufacturing: Unveiling Industry 5.0's human-centric evolution and digital twin revolution," *Heliyon*, vol. 10, no. 13, Art. no. e33551, 2024. https://doi.org/10.1016/j.heliyon.2024.e33551

[6] E. Traini, G. Antal, G. Bruno, M. De Maddis, F. Lombardi, L. Panza, and P. R. Spena, "Hybrid knowledge-based system supporting digital twins in Industry 5.0," *Procedia Computer Science*, vol. 232, pp. 1471–1480, 2024. https://doi.org/10.1016/j.procs.2024.01.145

[7] E. F. Langås, M. H. Zafar, and F. Sanfilippo, "Exploring the synergy of human–robot teaming, digital twins, and machine learning in Industry 5.0: A step towards sustainable manufacturing," *Journal of Intelligent Manufacturing*, vol. 37, no. 3, pp. 999–1022, 2026. https://doi.org/10.1007/s10845-025-02580-x

[8] I. Bucci, V. Fani, and R. Bandinelli, "Towards human-centric manufacturing: Exploring the role of human digital twins in Industry 5.0," *Sustainability*, vol. 17, no. 1, Art. no. 129, 2025. https://doi.org/10.3390/su17010129

[9] Z. Liu, J. Silva, R. Zhong, Q. Qin, N. Roy, V. N. Fernandez-Ayala, J. Lesko, U. Håkansson, S. Sandberg, D. V. Dimarogonas, J. Gross, X. V. Wang, and L. Wang, "ConstrucTwin: Digital twin-driven multirobot construction system toward Industry 5.0," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–16, 2026. https://doi.org/10.1109/TSMC.2026.3658622

[10] T. Coito, P. Faria, M. S. E. Martins, B. Firme, S. M. Vieira, J. Figueiredo, and J. M. C. Sousa, "Digital twin of a flexible manufacturing system for solutions preparation," *Automation*, vol. 3, no. 1, pp. 153–175, 2022. https://doi.org/10.3390/automation3010008

[11] A. Kovari, "A framework for integrating vision transformers with digital twins in the Industry 5.0 context," *Machines*, vol. 13, no. 1, Art. no. 36, 2025. https://doi.org/10.3390/machines13010036

[12] I. Chouridis, G. Mansour, A. Chouridis, V. Papageorgiou, M. T. Mansour, and A. Tsagaris, "Digital twin-driven four-dimensional path planning of collaborative robots for assembly tasks in Industry 5.0," *Robotics*, vol. 14, no. 7, Art. no. 97, 2025. https://doi.org/10.3390/robotics14070097

[13] S. A. K. Karanam and N. W. Hartman, “A systematic review of digital twin and virtual learning environments for smart manufacturing education,” *Manufacturing Letters*, vol. 44, pp. 1597–1608, 2025. https://doi.org/10.1016/j.mfglet.2025.06.179

[14] H. Asmat, I. U. Din, A. Almogren, and M. Y. Khan, “Digital twin with soft actor–critic reinforcement learning for transitioning from Industry 4.0 to Industry 5.0,” *IEEE Access*, pp. 1–17, 2025. https://doi.org/10.1109/ACCESS.2025.3546085

[15] A. Filipescu, G. Simion, A. Filipescu, and D. Ionescu, “IoT–cloud-based control of a mechatronic production line assisted by a dual cyber–physical robotic system within digital twin, AI, and Industry/Education 4.0/5.0 frameworks,” *Sensors*, vol. 26, no. 10, Art. no. 3194, 2026. https://doi.org/10.3390/s26103194

[16] M. Trstenjak, P. Gregurić, Ž. Janić, and D. Salaj, “Integrated multilevel production planning solution according to Industry 5.0 principles,” *Applied Sciences*, vol. 14, no. 1, Art. no. 160, 2024. https://doi.org/10.3390/app14010160

[17] A. Petrillo, A. Ferraro, A. Garofalo, and F. De Felice, “An industrial metaverse platform for collaborative and sustainable manufacturing in Industry 5.0,” *IET Collaborative Intelligent Manufacturing*, vol. 8, no. 1, Art. no. e70062, 2026. https://doi.org/10.1049/cim2.70062

[18] T. L. Leirmo, “Digital twins for Industry 5.0: Unlocking the human potential,” *Procedia CIRP*, vol. 130, pp. 761–766, 2024. https://doi.org/10.1016/j.procir.2024.10.161

[19] U. Asad, M. Khan, A. Khalid, and W. A. Lughmani, “Human-centric digital twins in industry: A comprehensive review of enabling technologies and implementation strategies,” *Sensors*, vol. 23, no. 8, Art. no. 3938, 2023. https://doi.org/10.3390/s23083938

[20] M. Q. Khan, M. A. H. Alvi, H. H. Nawaz, and M. Umar, “Impact of digital twins on real practices in manufacturing industries,” *Inventions*, vol. 10, no. 6, Art. no. 106, 2025. https://doi.org/10.3390/inventions10060106

[21] M. B. Al Zami, S. Shaon, V. K. Quy, and D. C. Nguyen, “Digital twin in industries: A comprehensive survey,” *IEEE Access*, pp. 1–46, 2025. https://doi.org/10.1109/ACCESS.2025.3551532

[22] V. Padmavathi, R. Kanimozhi, and R. Saminathan, “Digital twin-driven smart factories: Real-time physics-based co-simulation using edge AI and federated learning,” *Scientific Reports*, vol. 15, no. 1, pp. 1–14, 2025. https://doi.org/10.1038/s41598-025-28466-9

[23] Y. Wang, Q. Cui, M. Sun, and R. Han, “Research on the full lifecycle management of industrial mother machines empowered by human-centered intelligent manufacturing metaverse in the context of Industry 5.0,” in *Proceedings of the 2026 International Conference on Artificial Intelligence and Control*, pp. 823–833, 2026. https://doi.org/10.1145/3807246.3807374

[24] I. Rojek, D. Mikołajewski, E. Dostatni, J. Cybulski, and M. Kozielski, “Personalization of AI-based digital twins to optimize adaptation in industrial design and manufacturing,” *Applied Sciences*, vol. 15, no. 15, Art. no. 8525, 2025. https://doi.org/10.3390/app15158525

[25] M. De Marchi, A. Bonello, E. Francalanza, and E. Rauch, “A digital twin for SMEs in the context of Industry 5.0,” *Procedia CIRP*, vol. 126, pp. 242–247, 2024. https://doi.org/10.1016/j.procir.2024.08.333

[26] Z. Lei, J. Shi, Z. Luo, M. Cheng, and J. Wan, “Intelligent manufacturing from the perspective of Industry 5.0: Application review and prospects,” *IEEE Access*, vol. 12, pp. 167436–167451, 2024. https://doi.org/10.1109/ACCESS.2024.3496697

[27] M. Li, C. M. Yang, W. Lo, and Y. W. Kao, “A digital-twin-enabled AI-driven adaptive planning platform for sustainable and reliable manufacturing,” *Machines*, vol. 14, no. 2, Art. no. 197, 2026. https://doi.org/10.3390/machines14020197

[28] A. Lind, V. Elango, L. Hanson, D. Högberg, D. Lämkull, P. Mårtensson, and A. Syberfeldt, “Virtual-simulation-based multi-objective optimization of an assembly station in a battery production factory,” *Systems*, vol. 11, no. 8, Art. no. 395, 2023. https://doi.org/10.3390/systems11080395

[29] P. E. Dossou, E. Alvarez-de-los-Mozos, and P. Pawlewski, “A conceptual framework for optimizing performance in sustainable supply chain management and digital transformation towards Industry 5.0,” *Mathematics*, vol. 12, no. 17, Art. no. 2737, 2024. https://doi.org/10.3390/math12172737

[30] F. Mo, H. U. Rehman, J. C. Chaplin, D. Sanderson, and S. Ratchev, "Digital twin-based self-learning decision-making framework for industrial robots in manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 139, no. 1, pp. 221–240, 2025. https://doi.org/10.1007/s00170-025-15844-w

[31] E. Coronado, T. Ueshiba, and I. G. Ramirez-Alpizar, "A path to Industry 5.0 digital twins for human–robot collaboration by bridging NEP+ and ROS," *Robotics*, vol. 13, no. 2, Art. no. 28, 2024. https://doi.org/10.3390/robotics13020028

[32] S. M. M. Sajadieh and S. D. Noh, "A review of digital twin integration in circular manufacturing for sustainable industry transition," *Sustainability*, vol. 17, no. 16, Art. no. 7316, 2025. https://doi.org/10.3390/su17167316

[33] F. Zhou, K. Yu, W. Xie, J. Lyu, Z. Zheng, and S. Zhou, "Digital twin-enabled smart maritime logistics management in the context of Industry 5.0," *IEEE Access*, vol. 12, pp. 10920–10931, 2024. https://doi.org/10.1109/ACCESS.2024.3354838

[34] M. Crnjac Zizic, N. Gjeldum, M. Mladineo, B. Bilic, and A. Aljinovic Mestrovic, "Towards sustainable personalized assembly through human-centric digital twins," *Sensors*, vol. 25, no. 18, Art. no. 5662, 2025. https://doi.org/10.3390/s25185662

[35] F. G. Ciampi, T. M. Diallo, F. Mhenni, J. Y. Choley, and S. Patalano, "Digital twin-based framework for an efficient execution of CPPS reconfiguration through human–robot collaboration," *Journal of Manufacturing Systems*, vol. 83, pp. 157–174, 2025. https://doi.org/10.1016/j.jmsy.2025.09.001

[36] E. Novak, A. Javernik, I. Palčič, and R. Ojsteršek, "Developing a digital twin for human performance assessment in human–machine interaction," *Machines*, vol. 14, no. 3, Art. no. 346, 2026. https://doi.org/10.3390/machines14030346

[37] G. Grego, F. Nenna, and L. Gamberini, "Enhancing human–machine interactions: A novel framework for AR-based digital twin systems in industrial environments," in *Proceedings of the 17th International Conference on Pervasive Technologies Related to Assistive Environments*, pp. 456–462, 2024. https://doi.org/10.1145/3652037.3663946

[38] N. Tarek, A. D. Algarni, N. A. El-Hefnawy, H. Abdel-Kader, and A. Abdelatey, "Knowledge graph-enhanced digital twin framework for optimized job shop scheduling in smart manufacturing," *IEEE Access*, vol. 13, pp. 19863–19887, 2025. https://doi.org/10.1109/ACCESS.2025.3532600

[39] K. Djebbouri, H. Alofaysan, F. A. Hassan, and K. Si Mohammed, "Industry 5.0 digital DNA: A genetic code of human-centric smart manufacturing," *Sustainability*, vol. 17, no. 21, Art. no. 9450, 2025. https://doi.org/10.3390/su17219450

[40] A. Massaro, "Artificial intelligence signal control in electronic optocoupler circuits addressed on Industry 5.0 digital twin," *Electronics*, vol. 13, no. 22, Art. no. 4543, 2024. https://doi.org/10.3390/electronics13224543

[41] R. Patil and M. Löfstrand, "Robots and intelligent agents for product creation and manufacturing in sustainable digitalized manufacturing systems," *IOP Conference Series: Materials Science and Engineering*, vol. 1342, no. 1, Art. no. 012043, 2026. https://doi.org/10.1088/1757-899X/1342/1/012043

[42] S. Wang, J. Zhang, P. Wang, J. Law, R. Calinescu, and L. Mihaylova, "A deep learning-enhanced digital twin framework for improving safety and reliability in human–robot collaborative manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 85, Art. no. 102608, 2024. https://doi.org/10.1016/j.rcim.2023.102608

[43] C. Y. Wu, D. X. Zhu, M. Q. Huang, C. H. Hsu, and Z. J. Jia, "Actions and methods for achieving Industry 5.0-driven lean manufacturing transformation: A strategic roadmap," *Sustainability*, vol. 18, no. 12, Art. no. 6103, 2026. https://doi.org/10.3390/su18126103

[44] Y. Jia, W. Cheng, L. Meng, and C. Zhang, "Statistical learning-assisted evolutionary algorithm for digital twin-driven job shop scheduling with discrete operation sequence flexibility," *Symmetry*, vol. 17, no. 10, Art. no. 1614, 2025. https://doi.org/10.3390/sym17101614

[45] C. Turner and J. Oyekan, "Manufacturing in the age of human-centric and sustainable Industry 5.0: Application to holonic, flexible, reconfigurable, and smart manufacturing systems," *Sustainability*, vol. 15, no. 13, Art. no. 10169, 2023. https://doi.org/10.3390/su151310169

[46] P. Bernardin, Z. Hajicek, P. Janda, J. Kozak, F. Sedlacek, V. Lasova, and J. Kubicek, "Methodology of using CAx and digital twin methods in the development of a multifunctional portal centre in its pre-production phase," *Applied Sciences*, vol. 15, no. 6, Art. no. 3312, 2025. https://doi.org/10.3390/app15063312

[47] G. Mo, Y. Sun, S. Huang, F. Gao, J. Huang, L. Lai, G. Wang, and Y. Yan, "Reconfigurable digital twin from the perspective of smart manufacturing systems in Industry 4.0/5.0," *Journal of Physics: Conference Series*, vol. 2832, no. 1, Art. no. 012007, 2024. https://doi.org/10.1088/1742-6596/2832/1/012007

[48] P. Ding and J. Gao, "Digital twin- and AI-enabled intelligent optimisation design of agricultural machinery: A review," *Agronomy*, vol. 16, no. 11, Art. no. 1038, 2026. https://doi.org/10.3390/agronomy16111038

[49] K. Zidek, A. Nazim, A. Hosovsky, and J. Pitel, "Methodology of transfer virtual models to XR devices for human interaction in the concept of Industry 5.0," *IEEE Access*, pp. 1–13, 2026. https://doi.org/10.1109/ACCESS.2026.3691458

[50] Y. Jiang, W. Wang, J. Ding, X. Lu, and Y. Jing, "Leveraging digital twin technology for enhanced cybersecurity in cyber–physical production systems," *Future Internet*, vol. 16, no. 4, Art. no. 134, 2024. https://doi.org/10.3390/fi16040134

[51] V. Kuts, J. A. Marvel, M. Aksu, S. L. Pizzagalli, M. Sarkans, Y. Bondarenko, and T. Otto, "Digital twin as industrial robots manipulation validation tool," *Robotics*, vol. 11, no. 5, Art. no. 113, 2022. https://doi.org/10.3390/robotics11050113

[52] Z. Min, R. Shetty, S. Shekhar, A. D. Chhokra, M. Kritzler, T. Cui, A. Gupta, T. Ahlgrim, J. H. Reed, and A. Gokhale, "Challenges and opportunities in secure real-time digital twin systems for AR/VR-enabled smart factories," in *Proceedings of the International Workshop on Middleware for IT/OT Integration*, pp. 31–36, 2025. https://doi.org/10.1145/3774900.3776639

[53] Y. Qu, N. Zhao, and H. Zhang, "Digital twin technology of human–machine integration in cross-belt sorting system," *Chinese Journal of Mechanical Engineering*, vol. 37, no. 1, pp. 1–18, 2024. https://doi.org/10.1186/s10033-024-01012-w

[54] L. Bukowski and S. Werbinska-Wojciechowska, "Towards Maintenance 5.0: Resilience-based maintenance in AI-driven sustainable and human-centric industrial systems," *Sensors*, vol. 25, no. 16, Art. no. 5100, 2025. https://doi.org/10.3390/s25165100

[55] F. De Felice, A. Ferraro, L. Acampora, N. Sigh, and A. Petrillo, "Digital twin and metaverse technologies for sustainable manufacturing operations," *Procedia Computer Science*, vol. 274, pp. 444–452, 2025. https://doi.org/10.1016/j.procs.2025.12.044

[56] R. Anitha and A. Parthiban, "Smart waste ecosystems under Industry 5.0: A framework integrating digital twins, edge AI, graph theory, and 9R circularity," *Results in Engineering*, pp. 1–6, 2026. https://doi.org/10.1016/j.procir.2026.01.010

[57] D. Daraba, F. Pop, and C. Daraba, "Digital twin used in real-time monitoring of operations performed on CNC technological equipment," *Applied Sciences*, vol. 14, no. 22, Art. no. 10088, 2024. https://doi.org/10.3390/app142210088

[58] N. Al-Qirim, A. Bani-Hani, M. Majdalawieh, H. Al Hamadi, and M. K. Hasan, "DiGraph-enabled digital twin and label-encoding machine learning for SCADA network cyberattack analysis in Industry 5.0," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 3404–3416, 2025. https://doi.org/10.1109/OJCOMS.2024.3502544

[59] Z. Ji, X. Fu, and L. Zhao, "Automated deployment scheme of digital twin manufacturing system based on topology optimization and multiscale modeling," *Modelling and Simulation in Engineering*, vol. 2026, no. 1, Art. no. 4344492, 2026. https://doi.org/10.1155/mse/4344492

[60] M. A. Hazrat, N. M. S. Hassan, A. A. Chowdhury, M. G. Rasul, and B. A. Taylor, "Developing a skilled workforce for future industry demand: The potential of digital twin-based teaching and learning practices in engineering education," *Sustainability*, vol. 15, no. 23, Art. no. 16433, 2023. https://doi.org/10.3390/su152316433