Politecnico di Torino

Master of science program

In Engineering and Management

A.a. 2022/2023

graduation session 2023

Master of science program:

Reinforcement learning applications in manufacturing.

Advisor:

Giulia Bruno

Candidate:

Sonia Belli

1. Summary

2. Abstract

In the landscape of modern manufacturing, Reinforcement Learning (RL) stands out as a promising frontier, offering transformative solutions to different challenges. This thesis embarks on a comprehensive exploration of RL applications in manufacturing, seeking to unravel the potential of this machine learning paradigm in optimizing diverse processes. Manufacturing, with its multifaceted operations, demands intelligent approaches for efficient decision-making, resource allocation, and system performance. The aim of this research is to bridge the theoretical understanding of RL with practical implementations, providing nuanced insights into how RL can revolutionize manufacturing practices.

This study's primary objective is to conduct an extensive examination of RL applications across various facets of manufacturing. A rigorous literature review sets the stage for practical experiments, aiming to evaluate RL's efficacy in addressing contemporary challenges within manufacturing environments. By delving into real-world applications, this research aspires to not only contribute theoretical knowledge but also to provide actionable insights for practitioners and decision-makers in the manufacturing domain.

3. Introduction

Manufacturing processes are undergoing significant change because of the fourth industrial revolution, or "Industry 4.0”. Additionally, there have been significant changes in how people and machines interact in the industrial sector, leading to the idea of "Industry 5.0”. The conventional approaches to manufacturing are undergoing significant transformations due to the digitization of enterprises and production facilities. This transformation is characterized by the integration of machines through embedded systems and the Internet of Things (IoT), the emergence of collaborative robots (cobots), the utilization of individual workstations, and the implementation of matrix production. There is a growing need for personalized and customized products in the market. In response, there is a surge in the number of orders coupled with a decrease in batch sizes, reaching the extent of fully decentralized 'batch size one' production. The demand for a high degree of diversity in production is inevitable due to the rise of Mass Customization. This approach to manufacturing requires processes that are highly adaptable and flexible.

Machine Learning (ML) plays a crucial role in making production more intelligent, providing the necessary capabilities for increased flexibility and adaptability. These advancements in machine learning are driving the era of smart manufacturing, often referred to as Industry 4.0. Consequently, machine learning is gaining growing importance in the manufacturing sector, alongside digital solutions and sophisticated technologies like the Industrial Internet of Things (IIoT), additive manufacturing, digital twins, advanced robotics, cloud computing, and augmented/virtual reality.ML is an Artificial Intelligence (AI) area that covers algorithms that learn directly from their input data.

The goal of using ML in manufacturing is to accomplish production optimization at four separate levels: product, process, machine, and system. As a result, the use cases for applying ML may be further classified by these distinct levels, as illustrated in Figure 1, of the ML typical use.

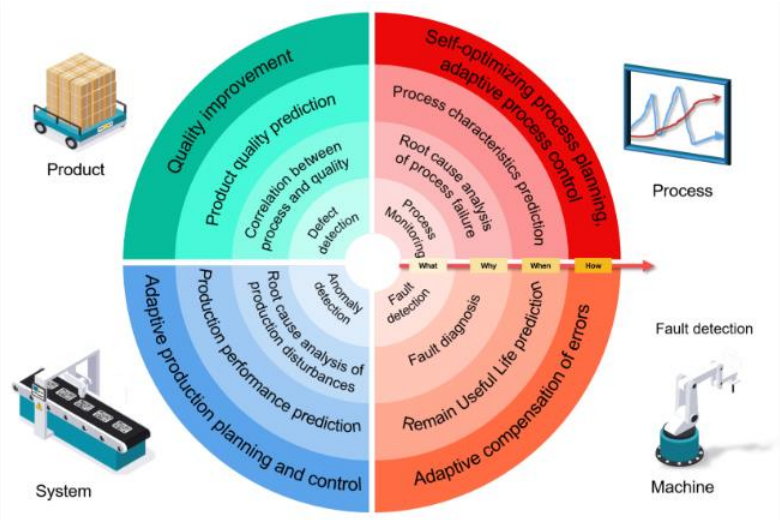

Figure 1. RL levels in manufacturing applications [57]

4. Machine Learning Introduction

Recently, a paradigm change, in a variety of global businesses, from technology to health care, has been brought about by artificial intelligence (AI). A vast number of industrial and academic brains are being ignited by the once obscure subject. The capacity of AI to "self-learn" in combination with the quick development of computer technology and the decreasing cost of data storage has propelled AI to the forefront of algorithms for many applications, including computer vision and natural language processing. By 2030, AI is expected to contribute over $15 trillion USD to the global economy and increase GDP by 26%, according to PwC (2019). Overall, AI is a vast area with several objectives. Currently, the most influential topic in AI is machine learning (ML). ML can be described as the scientific field that studies and develops algorithms and statistical models to give machines the explicit ability to learn tasks without being programmed to do so (Russel and Norvig, 2009). The ML field can be further decomposed into supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning [57].

4.1. Supervised learning

Supervised learning is a task-oriented method, involving the process of a machine learning a function that transforms an input into an output based on examples of input-output pairs. This learning approach requires labeled training data, consisting of a set of training examples. Supervised learning

is employed when specific objectives need to be achieved from a defined set of inputs.In particular, supervised learning methods strive to learn an approximation function, denoted as f, capable of mapping inputs x to outputs y with the guidance of annotations such as $(x1,y1),(x2,y2),\ldots,(xN,yN)$. In this process, the algorithm analyses a labelled dataset and generates an inferred function that can be applied to unseen samples.

It's essential to highlight that supervised learning relies on labelled datasets, making it imperative to have a significant amount of data and incurring high labelling costs. This learning method is commonly utilized for addressing two primary problems: regression and classification. The distinction lies in the data type of the output variables, where regression predicts continuous numeric values ($y \in \mathbb{R}$), while classification predicts categorical values ($y \in \{0,1\}$).In terms of principles, supervised learning methods can be further categorized into four groups: tree-based methods, probabilistic-based methods, kernel-based methods, and neural network-based methods [8].

- **Tree-based approaches:** involve dividing the feature space into distinct areas, ensuring that data points within each region share a similar class or value. This process results in the creation of a tree-like structure with if-then rules, which can be employed to determine the target class or value. Unlike some black-box models used in other supervised methods, tree-based approaches offer enhanced comprehensibility and higher model interpretability. The key advantage of tree-based approaches lies in their ability to provide clear insights into the decision-making process, making them more interpretable compared to other complex models. This characteristic is particularly valuable in scenarios where understanding the reasoning behind predictions is crucial. In the realm of manufacturing, especially at the product and machine level, tree-based approaches find applications in identifying influencing factors leading to quality defects or machine failures. By leveraging their interpretability, these approaches enable effective problem diagnosis and contribute to a deeper understanding of the factors influencing outcomes in the manufacturing process. In addition, the identified important factors can help in further predicting target values such as product quality events of interest before they happen, such as machine breakdown [8].

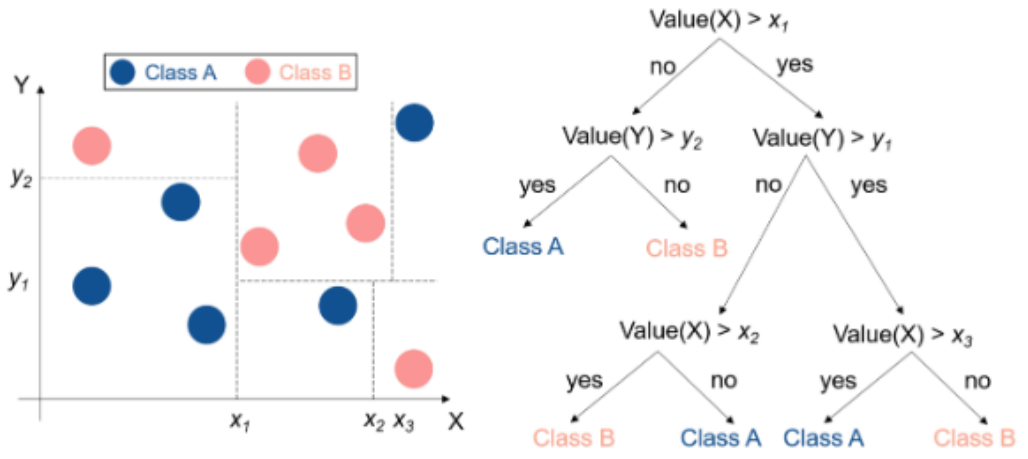

Figure 2. The principle of a decision tree [8].

- Probabilistic-based methods: such as Bayesian Optimization (BO) and Hidden Markov Models (HMM), offer a different approach to modeling by providing probabilities for each class as the output. These models are adept at handling and explaining the inherent uncertainties present in data, allowing for the construction of hierarchically complex models. Bayesian networks, a type of probabilistic model, excel in capturing dependencies among different variables. This capability is particularly advantageous in manufacturing applications, where the detection or prediction of events like quality issues, machine failure, or dynamic process modelling involves understanding intricate relationships between variables. Markov chains, another probabilistic model type, describe sequences of possible events, where the probability of each event depends solely on the state achieved in the preceding event. This sequential modelling approach is valuable in scenarios where the evolution of a system depends on its recent history, making it suitable for applications requiring the prediction of sequential events in manufacturing processes. Markov chains can be utilized in manufacturing to model and analyse the behaviour of systems such as production lines or supply chains. In addition, the capability of predicting future states with Markov chains enables applications predicting joint maintenance in production systems and optimizing production scheduling [8].

- Kernel-based methods: as illustrated in Figure 4, leverage a designated kernel function to transform input data into a high-dimensional implicit feature space. Instead of explicitly calculating the targeted coordinates, these methods typically compute the inner product between pairs of data points within the feature space. It's worth noting that kernel-based methods may face efficiency challenges, especially when dealing with large-scale input data. Despite this, they exhibit promising capabilities in classification and regression tasks, making them valuable for manufacturing applications like defect detection, quality prediction, and wear prediction in machinery.
Supervised learning encompasses various types of kernel-based methods, including Support Vector Machines (SVM) and Kernel–Fisher Discriminant Analysis (KFD). These approaches contribute to enhancing the understanding and prediction of complex relationships within manufacturing processes.

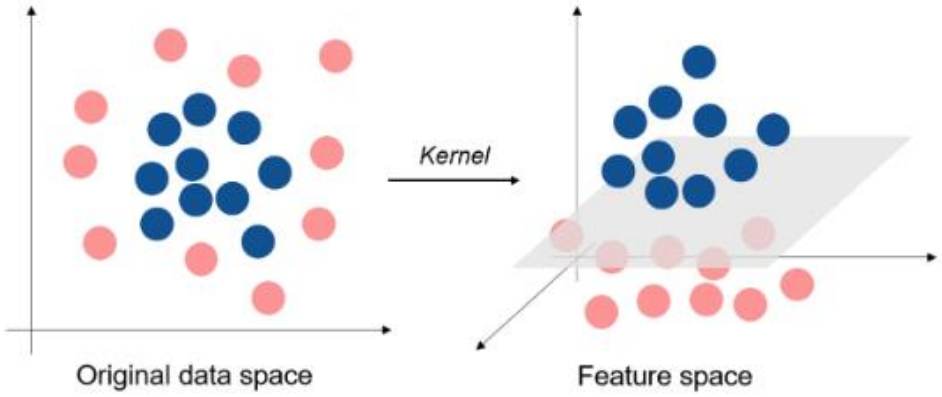

Figure 3. The principle of kernel-based methods. Using a kernel, the linearly inseparable input data are transformed to another feature space in which they become linearly separable. [8]

- **Neural-network-based methods**: Taking inspiration from biological neurons and their intercellular communication, neural network-based methods leverage artificial neurons. A typical neural network, including Artificial Neural Networks (ANNs), comprises an input layer, hidden layer, and output layer, as depicted in Figure 4. Various types of ANNs, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Deep Belief Networks (DBNs), play pivotal roles in extracting meaningful features from different types of data.CNNs, renowned for their adept feature extraction from matrix-like data, find widespread application in image processing. In the manufacturing domain, CNNs excel in tasks like image-based quality control and process monitoring. Moreover, by transforming sensor-generated time series data into 2D images, CNNs can contribute to detecting and diagnosing machine failures.RNNs, tailored for processing sequential input data such as time series or sequential images, are well-suited for analyzing sensor data or live machine images in manufacturing applications. They enable real-time performance predictions, including forecasting the remaining useful life of machinery, predicting process behavior, or forecasting production indicators crucial for real-time production scheduling.

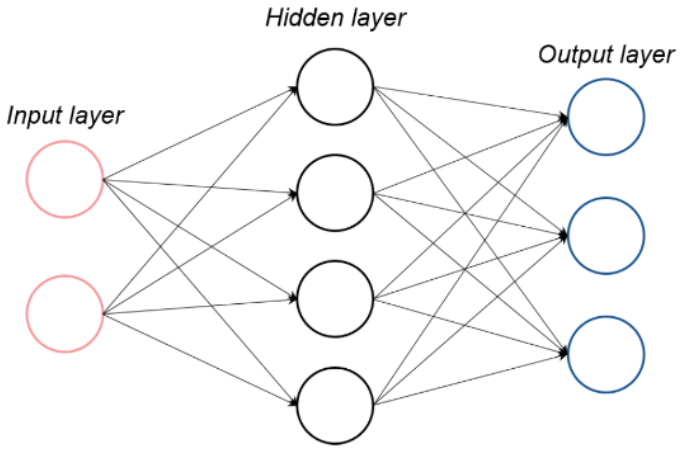

Figure 4. The scheme of an ANN, which normally consists of an input layer, hidden layer and output layer. [8].

4.2. Unsupervised learning

Unsupervised learning is a data-driven approach that explores unlabelled datasets, allowing algorithms to learn from the data without predefined outputs or target variables. This methodology is often applied for generative feature extraction, identifying relevant trends and structures, grouping results, and experimental purposes. The primary objective of unsupervised learning is to uncover hidden and meaningful patterns within unlabelled data. It encompasses three fundamental types of unsupervised tasks: Dimension Reduction, Clustering, and Association Rules. Unsupervised learning holds significant potential in various manufacturing applications. Clustering algorithms, for instance, can effectively identify outliers in manufacturing data. Moreover, in scenarios involving high-dimensional data, such as manufacturing cost estimation, quality improvement strategies, production process optimization, and customer data analysis, unsupervised learning methods prove valuable. Dealing with the complexity and high dimensionality of data often requires the assistance of

dimensional reduction support algorithms. Finally, when conducting root cause analysis in large-scale process executions, especially in complex data center services, association rule-based learning becomes instrumental in identifying correlations between variables within a dataset.

- Semi-supervised learning: it combines aspects of both supervised and unsupervised approaches, utilizing both labelled and unlabelled data. This approach falls between learning "without supervision" and learning "with supervision." Semi-supervised learning is particularly valuable in real-world scenarios where unlabelled data is abundant, but labelled data is scarce. The primary goal of a semi-supervised learning model is to generate predictions that outperform those made solely with the available labelled data. This approach finds applications in various fields, including text categorization, machine translation, and fraud detection. Semi-supervised learning methods can be broadly categorized into two groups: data augmentation-based methods and semi-supervised mechanism-based methods.

- Data augmentation: by leveraging data augmentation, labelled datasets can be expanded and enriched by incorporating model predictions from newly acquired unlabelled data, particularly those with high confidence as pseudo-labels [8]. Data augmentation procedures are straightforward, and there's no requirement for meticulous loss design. Consequently, data augmentation-based strategies for augmenting labelled datasets hold potential utility for non-experts in manufacturing, especially in scenarios where large volumes of unlabelled data are readily available.

- Semi-supervised mechanisms: In contrast, semi-supervised mechanism-based methods concentrate on the process of utilizing both labelled and unlabelled data. Here, both labelled and unlabelled data can serve as inputs to the model, and their losses are computed in distinct manners. Examples of applications in manufacturing include quality monitoring based on images, process fault detection, and anomaly detection in machinery.

4.3. Reinforcement learning

It is also known as an environment-driven technique, is a form of machine learning algorithm that enables software agents and machines to automatically assess the ideal behaviour in a specific context or environment to increase its efficiency. The goal of this incentive-based or penalty-based learning approach is to use the knowledge gained from environmental activists to take steps that will either maximise the benefit or minimise the risk. However, it is not recommended to use it for resolving simple or elementary issues. It is a strong tool for training AI models that can help increase automation or optimize the operational efficiency of complex systems like robotics, autonomous driving tasks, manufacturing, and supply chain logistics.

As a result, depending on the nature of the data stated earlier and the desired result, various machine learning techniques can play a key role in the development of effective models in a variety of application areas.

5. Basics of Reinforcement Learning

The learner or decision-maker is known as the **agent** in RL literature, and the setting in which the agent exists and interacts is known as the **environment**. The agent can interact with the environment by taking certain **actions**, but such activities have no impact on the dynamics or laws of the environment. In RL literature, the environment's current state is referred to as the **state**, and RL agents take actions based on the state and **reward** signals. To teach RL, both rewards and sanctions are used.

Iterative learning is the foundation of reinforcement learning algorithms. Trial and error, as well as the interaction of an agent with its environment, are the foundations of learning. A Markov Decision Process (MDP) is used to model this interaction. This concept reduces the interaction to three signals:

- State s: the environment's current situation.
- Action a: agent's operation or decision based on the state and its experience.
- Reward r: represented by the environment's numerical feedback. It teaches the agent and let it know whether its action was successful or not.

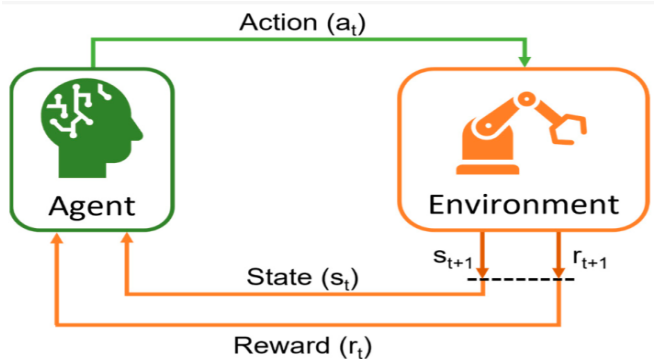

Figure 5. Structure of Markov Decision Process. [13]

The agent's goal is to maximize cumulative reward over time, which requires mastering the job. This is mirrored in the agent's policy, which specifies which course of action is preferable in each stage. As the agent interacts with the environment and acquires experience, this policy is updated and enhanced.

Reinforcement learning (RL) has always possessed a remarkable capacity for sequential decision-making. The past ten years have seen a rapid increase in high-performance computing power and the development of deep learning (DL) techniques. As a result, algorithms combining RL with deep neural networks, or DRL, have not only improved environment perception but also allowed RL algorithms to perform better, adapt, and make decisions more quickly. DRL has received a lot of attention recently from the industry field in addition to being widely used in games, banking, network communication systems and robot control.
As the manufacturing paradigm shifts toward mass personalisation manufacturing systems need to respond to orders with a shorter lead-time and higher quality, which necessitates the production process being more flexible and adaptable.

Due to its self-learning skills to make precise and quick judgements in dynamic and complicated scenarios, DRL offers significant potential in these circumstances.

Unlike other supervised learning applications (such as computer vision or natural language processing, etc.), DRL uses trial-and-error techniques to self-optimize by interacting with the environment without using any manually labelled data. DRL's self-learning capabilities and labelled data features greatly reduce the need for human intervention and make it simple to adopt and implement. Meanwhile, DRL's capacity for quick judgment and extrapolation from prior knowledge in the face of challenging circumstances are demonstrated by DeepMind's AlphaGo series application. Researchers are now aware of the benefit in engineering disciplines including autonomous driving, the internet of things, and robot systems, and in-depth reviews are provided.

Despite the DRL applications in smart manufacturing expanding rapidly, there isn't yet enough research to fully explain the state of the field and highlight unresolved problems. Therefore, a state-of-the-art review of manufacturing DRL's applications which analyse current trends, crucial challenges and limitations must be carried out.

6. Paper Selection

The fundamental steps of the literature evaluation process for DRL applications in smart manufacturing applications are described in this paragraph. One well-known academic database, Scopus, was mostly used for the literature search as it includes many peer-reviewed, interdisciplinary research publications, indeed many studies on DRL could be found.

The terms "manufacturing", "production", "reinforcement learning" were used as keywords and the period covered by the review was 2013–2023, and only English-language literature was considered. "Reinforcement learning" was used as the keyword to search for relevant literature, even though another focal algorithm type in the review is DRL, because during the early stages of DRL, researchers did not properly distinguish between the names DRL and RL. 2013 was chosen as the starting year of reviews as it was the year when the representative DRL study, received a lot of attention.

Therefore, the main research question was formulated to start the paper selection: What are the main (deep) reinforcement learning applications in manufacturing processes? Then, this question was translated in a search through this engine. The search was carried out using the keywords listed above and logic operators that delimited the search field.

The search was performed with the following query:

- TITLE-ABS-KEY (("manufacturing " OR " production ") AND " reinforcement learning ") AND PUBYEAR > 2012 AND PUBYEAR < 2024 AND PUBYEAR > 2012 AND PUBYEAR < 2024 AND (LIMIT-TO (LANGUAGE , "English"))

From the previous query, a total of 2051 documents were returned. For a high-quality review, the scope was narrowed down considering only papers that have as 'Subject area: Engineering'. The result of this added filter was a total of 1260 documents. To obtain a manageable number of papers to

review, without losing consistency in the research, the ‘Search within’ section of Scopus was changed to ‘Article title’. The following query finally was carried out:

> TITLE ((“manufacturing " OR " production ") AND " reinforcement learning ") AND PUBYEAR > 2012 AND PUBYEAR < 2024 AND PUBYEAR > 2012 AND PUBYEAR < 2024 AND (LIMIT-TO (LANGUAGE , "English")) AND (LIMIT-TO (SUBJAREA , "ENGI"))

The query returned a total of 190 papers and the ‘Analyze’ function of the research tool was used to classify the documents by year, author, type and country as shown below.

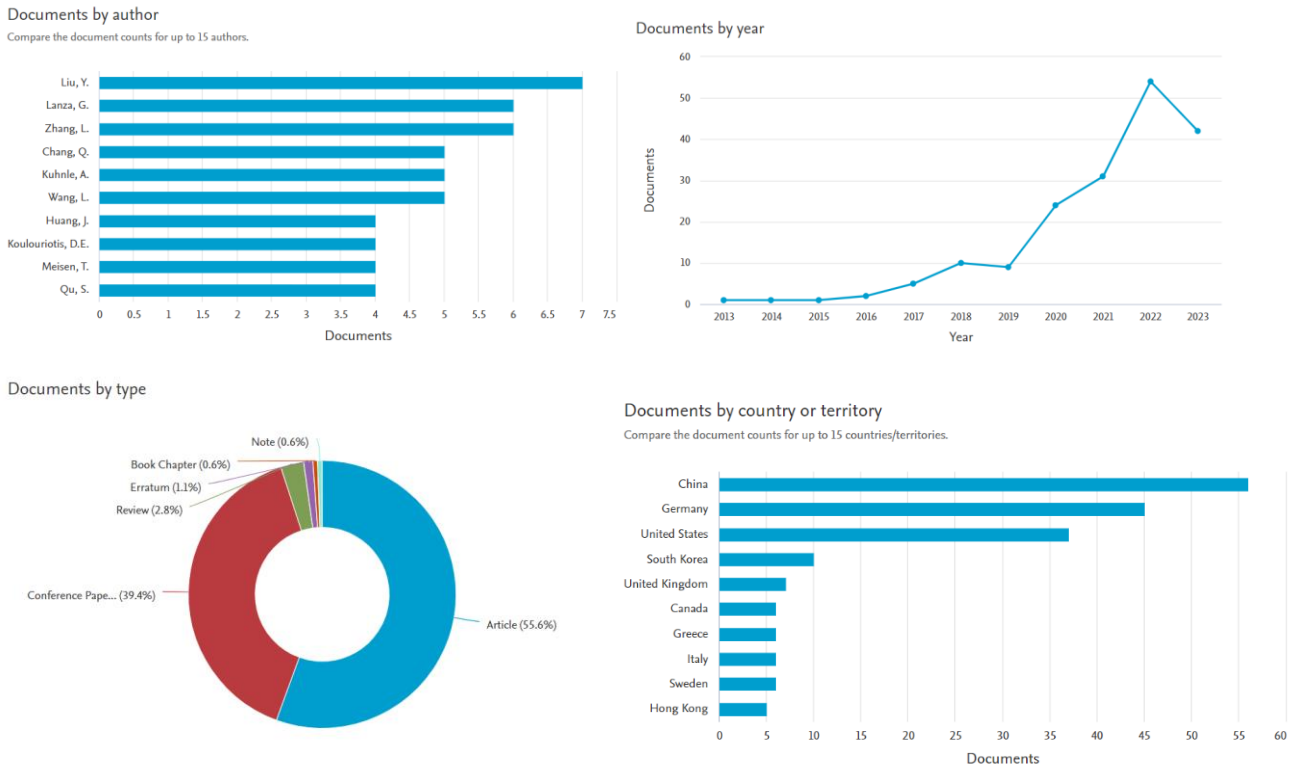

Figure 6. Research documents' classification

To restrict the scope of the research a little bit more I’ve filtered out work that did not fit the objective domain or that resulted out of scope based on title, abstract or article browsing.
The selected documents are listed in table 1.

6.1. Selected Papers

TITLE	AREA	SUB-AREA	TOPIC	METHOD USED	DATASET
User-guided motion planning with reinforcement learning for human-robot collaboration in smart manufacturing [79]	Motion Planning	The goal is to develop a scalable and adaptive motion planning method to automatically generate motion plans for new robotic manipulation tasks without manually reprogramming robots	development of a scalable and adaptive motion planning method to automatically generate motion plans for new robotic manipulation tasks without manually reprogramming robots	Q-learning algorithm	private simulated and real data
Spatiotemporal path tracking via deep reinforcement learning of robot for manufacturing internal logistics [17]	Motion Planning	To improve the logistics ability of robots in real industrial scenarios, the paper proposes a spatiotemporal path tracking control system (multi-scenario and multi-stage training) based on RL	definition of a logistics system based on cloth-roll handling robot (CHR) and its DRL path tracking system in weaving workshop	Dynamic Observing Markov Decision Process (DOMDP) algorithm	private real and simulated data
Simulation and deep reinforcement learning for adaptive dispatching in semiconductor manufacturing systems [72]	Scheduling	RL application for dispatching and resources allocation	DQN-based dispatching and resources allocation approach for a semiconductor manufacturing system	Deep-Q-Network (DQN)	private simulated data
Sequence generation for multi-task scheduling in cloud manufacturing with deep reinforcement learning [66]	Scheduling	cloud manufacturing scheduling (CMfg-Sch) problem	development of a DQN- and Double DQN-based multi-task scheduling algorithms and application of them on a case study for the production and assembly of a crankshaft flywheel set	DQN- and Double DQN-based multi-task scheduling algorithms.	private simulated data
Scheduling of decentralized robot services in cloud	Scheduling	Robotics and Computer-Integrated	development of DQN- and DDQN-based scheduling algorithms	DQN- and DDQN-based	private simulated data

manufacturing with deep reinforcement learning [49]		Manufacturing scheduling problem	to manage and schedule decentralized robot services in cloud manufacturing to achieve on-demand provisioning	scheduling algorithms	
Reinforcement learning-based dynamic production-logistics-integrated tasks allocation in smart factories [41]	Scheduling	production-logistics-integrated tasks allocation problem	Deployment and simulation of a RL algorithm to allocate production and logistics tasks in SFs co-ordinately and autonomously.	Q-learning algorithm	private simulated data
Reinforcement learning for process control with application in semiconductor manufacturing [45]	Process control	Process control problem	Development and simulation of RL-based controllers (with or without domain knowledge) in linear and nonlinear simulation cases	RL-based controller with approximate models and RL-based controller with Policy Gradient Search (PGS)	private simulated data
Reinforcement Learning Enabled Autonomous Manufacturing Using Transfer Learning and Probabilistic Reward Modelling [50]	Autonomous Manufacturing	Autonomous Manufacturing problems in complex geometries industry	implementation of a RL algorithm in high variable cost environments such as autonomous manufacturing systems that can learn the manufacturing process parameters to autonomously fabricate a complex geometry artifact with desired performance characteristics	off-policy random sample Q-learning	private simulated data
Reinforcement learning based trustworthy recommendation model for digital twin-driven decision-support in	Scheduling	Production and logistics task allocation problem	development of an innovative digital twin decision support framework that integrates recommendation	Q-learning algorithm	private simulated and real data

manufacturing systems [67]			systems with the RL algorithm		
Reinforcement learning and optimization-based path planning for thin-walled structures in wire arc additive manufacturing [65]	Autonomous Manufacturing	path planning and process optimization in AM	development of a path planning framework named RLPlanner to enable a fully automatic deposition path planning for thin-walled structures in wire arc additive manufacturing	Proximal Policy Optimization (PPO)	private simulated data
Post-prognostics demand management, production, spare parts, and maintenance planning for a single-machine system using Reinforcement Learning [88]	Maintenance Strategies and Quality	Maintenance problem to improve Production Planning and Control (PPC)	a data-driven post-prognostics RL model was developed that improves and automates Production Planning and Control decision-making	Deep Q-Learning (DQL), a Proximal Policy Optimisation (PPO) and an Advantage Actor Critic (A2C) algorithm.	private simulated data
Multi-objective reinforcement learning-based framework for solving selective maintenance problems in reconfigurable cyber-physical manufacturing systems [4]	Maintenance Strategies and Quality	selective maintenance problems	development and simulation of a robust model for a selective maintenance problem with imperfect repairs in the reconfigurable cyber-physical systems (RCPMS) context	Deep Q Network (DQN), multi-objective reinforcement learning (MORL)	private simulated data
Multi-agent deep reinforcement learning for task offloading in group distributed manufacturing systems [91]	Scheduling	task offloading in group distributed manufacturing systems	a MaDRLAM with attention mechanism is proposed to solve the task offloading problem in distributed manufacturing systems	Multi-agent deep reinforcement learning (MaDRLAM)	private simulated data
Logistics-involved task scheduling in cloud manufacturing with offline DRL[84]	Scheduling	cloud manufacturing scheduling problems (CMfg-SPs)	definition of an offline DRL scheduling algorithm to address CMfg-SPs	Markov decision process (MDP)	private simulated data

Joint optimization of maintenance and quality inspection for manufacturing networks based on deep reinforcement learning [93]	Maintenance Strategies and Quality	the MDP-based optimization model, the proposed Deep Deterministic Policy Gradient (DDPG) algorithm realizes the optimal reliability-quality joint control in manufacturing networks.	joint optimization problem of preventive maintenance and work-in-process quality inspection for manufacturing networks with reliability-quality interactions.	Deep Deterministic Policy Gradient (DDPG) algorithm	private simulated data
Inverse Reinforcement Learning Framework for Transferring Task Sequencing Policies from Humans to Robots in Manufacturing Applications [58]	Scheduling	task sequencing for robots in complex manufacturing processes.	development and implementation of a learning task sequencing policy based on inverse reinforcement learning (IRL)	Performance-based Preference Learner and Effort-based Preference Learner	private real and simulated data
Graph neural networks-based scheduler for production planning problems using reinforcement learning [21]	Scheduling	job shop scheduling problems (JSSP)	designation of a novel framework named GraSP-RL, GRAph neural network-based Scheduler for Production planning problems using RL	Proximal Policy Optimization (PPO)	private simulated data
Explainable multi-agent deep reinforcement learning for real-time demand response towards sustainable manufacturing [94]	Sustainable Manufacturing	energy management and demand response for sustainable industrial development.	use of RL to control a section of an automotive assembly line using one year of DR (demand response) electricity price data to validate its performance	decomposed multi-agent deep Q-network (DMADQN)	private simulated data
Dynamic scheduling for semiconductor manufacturing systems with uncertainties using convolutional neural networks and reinforcement learning [48]	Scheduling	Production scheduling problem of semiconductor manufacturing systems (SMSs)	The paper studied the dynamic release control and production scheduling problem of SMSs while considering uncertainties from the internal and external environment. The proposed a CNN-A3C-based approach is evaluated is on the semiconductor	convolutional neural network (CNN)- and asynchronous advanced actor critic (A3C)-based method	private simulated data

			smart manufacturing demonstration unit system that the research group established according to the benchmark Minifab.	called CNN-A3C	
Dynamic production scheduling towards self-organizing mass personalization: A multi-agent duelling deep reinforcement learning approach [69]	Scheduling	production scheduling	deployment of a dynamic scheduling system	Multi-Agent Duelling DQN	private simulated data
Dynamic Maintenance for a Large Scale Identical Parallel Manufacturing Systems Using Reinforcement Learning [52]	Maintenance Strategies and Quality	maintenance decision making for cost minimization	(RL) approach for dynamic maintenance model for multi-component parallel system subject to stochastic degradation and random failures	Q-learning algorithm	private simulated data
Distributed Real-Time Scheduling in Cloud Manufacturing by Deep Reinforcement Learning [37]	Scheduling	distributed real-time scheduling problem of processing services with logistics constraints in CM	Implementation of a D3QN with cloud edge collaboration for distributed real-time scheduling of processing and logistics services and its validated in dynamic job shop scheduling problems	distributed duelling deep Q network (D3QN) with cloud–edge collaboration	private simulated data
Design and Implementation of Simulation-Based Scheduling System with Reinforcement Learning for Re-Entrant Production Lines [28]	Scheduling	manufacturing scheduling problem	Design and Implementation of a Simulation-Based Scheduling System with RL and its evaluation on a hypothetical re-entrant production line.	double deep Q-network (DDQN) algorithm	private simulated data
Demand Response Optimization of Cement Manufacturing Industry Based on Reinforcement Learning Algorithm [89]	Sustainable Manufacturing	industrial demand response	Modelling and analysis of a complete industrial demand response scheduling framework based on Markov	Network-Based Proximal Policy Optimization (PPO)	private simulated data

			decision process in the cement industry.		
Cloud–edge collaboration task scheduling in cloud manufacturing: An attention-based deep reinforcement learning approach [9]	Scheduling	cloud manufacturing scheduling (CMfg-Sch) problems	application of attention-based DRL framework to solve the CMfg task scheduling problem of cloud–edge collaboration	AV-MPO, SAC, PPO, V-MPO, and Duelling DQN	private simulated data
Application of a Reinforcement Learning-based Automated Order Release in Production [73]	Scheduling	Order release in the job shop scheduling problem (JSP)	Elaboration on the usage of reinforcement learning algorithms for automated order release in a practice-based application	DQN algorithm	private simulated data
An improved deep reinforcement learning-based scheduling approach for dynamic task scheduling in cloud manufacturing [83]	Scheduling	Dynamic task scheduling problem in cloud manufacturing (CMfg)	This paper proposes an improved DRL-based scheduling algorithm for the DTSP-CMfg in the automotive sector	proximal policy optimizati on (PPO)	private simulated data
A reinforcement learning/ad-hoc planning and scheduling mechanism for flexible and sustainable manufacturing systems [63]	Sustainable Manufacturing	Process scheduling in the manufacturing industry	Optimization of failure-prone machines integrated in a multi-stage production line, processing one type of products using RL	model-free average reward algorithms	private simulated data
A reinforcement learning approach for process parameter optimization in AM [14]	Process control	process parameter optimization in AM	process parameter optimization for melt pool depth of a AM system i.e., powder-fed L-DED.	model-free, off-policy Q-Learning	private simulated data
A Reinforcement Learning Algorithm for Optimal Dynamic Policies of Joint Condition-based Maintenance and Condition-based Production [19]	Maintenance Strategies and Quality	joint condition-based maintenance and production	development of joint optimal maintenance and production policy based on MDP for a specific type of production system that allows for adjustable production rates	Markov decision process (MDP)	private simulated data
A multi-objective reinforcement learning approach for resequencing scheduling	Scheduling	resequencing scheduling problem	Investigation of a multi-objective resequencing scheduling problem in the automotive	MORL-based Multi-Objective-	private simulated data

problems in automotive manufacturing systems [42]			manufacturing systems (operational requirements on the colour-batching of the paint shop and sequential requirement)	Deep-Q-Network (MODQN)	
Distributed Real-Time Scheduling in Cloud Manufacturing by Deep Reinforcement Learning [37]	Scheduling	production-maintenance joint scheduling task of a production system	development and simulation (using Digital twin) of a DQN algorithm for job scheduling and production equipment maintenance	Deep Q Network (DQN)	private simulated data
Design and Implementation of Simulation-Based Scheduling System with Reinforcement Learning for Re-Entrant Production Lines [28]	Scheduling	Production Planning and control (PPC)	To improve the dynamic responsiveness and production efficiency of manufacturing workshop to personalized orders, a multiagent manufacturing system with the ability of online scheduling and scheduling strategy optimization is constructed and implemented using a digital workshop.	proximal policy optimization (PPO) algorithm and deep Q network (DQN)	private simulated data
Reinforcement learning-based defect mitigation for quality assurance of additive manufacturing [11]	Maintenance Strategies and Quality	quality assurance in additive manufacturing (AM)	development and simulation of online learning-based method (CGL) to deal with the new defects during printing in AM. The proposed method addresses the challenge of limited samples in AM process by transferring offline and online prior knowledge into the current AM process.	Continual G-learning	private simulated data

Solving task scheduling problems in cloud manufacturing via attention mechanism and deep reinforcement learning [86]	Scheduling	task scheduling problems in CMfg	proposition of an end-to-end scheduling algorithm to address the CMfg-SP through the attention mechanism and DRL to maximize the quality of service (QoS)	Markov decision process (MDP) in particular the REINFORCE algorithm	private simulated data
Hybrid feedback and reinforcement learning-based control of machine cycle time for a multi-stage production system [43]	Process control	scheduling and control of machines' operations (to increase system's efficiency)	development of a control method for multi-stage production systems to dynamically change the individual machines' cycle time to improve overall system efficiency.	Standard and Extended advantage actor critic (A2C)	private simulated data
Dynamic scheduling of tasks in cloud manufacturing with multi-agent reinforcement learning [85]	Scheduling	cloud manufacturing scheduling (CMfg-Sch) problems	development of a MAGCIS algorithm to solve the GSCMfg scheduling problem. Simulation of the algorithm in the processing of aircraft structural parts and performance comparisons with other RL algorithms	multi-agent graph convolution integrated scheduler (MAGCIS)	private simulated data
Reinforcement Learning Enabled Self-Homing of Industrial Robotic Manipulators in Manufacturing [31]	Motion Planning	self-homing (HPos) problems in industrial manufacturing robots	development and simulation of a SAC algorithm in brazing and assembly applications for aircraft engines to solve the home position problem of industrial robotic manipulators	Soft Actor-Critic (SAC)	private simulated data
Using real-time manufacturing data to schedule a smart factory via reinforcement learning [18]	Scheduling	real time scheduling problem in smart factory	To realize the data-driven manufacturing, this paper proposes the cyber-physical architecture for smart factory, and uses CNP to design the MAS-	double Q-learning	private simulated data

			based dynamic scheduling mechanism		
Multi-Agent Reinforcement Learning for Real-Time Dynamic Production Scheduling in a Robot Assembly Cell [12]	Scheduling	dynamic flexible job shop scheduling (FJSP) in a robot assembly cell	Multi-Agent Reinforcement Learning solution based on Double DQN for a dynamic FJSP setting in a robot assembly cell.	Double DQN-based algorithm	private simulated data
Graph neural network and multi-agent reinforcement learning for machine-process-system integrated control to optimize production yield [25]	Process control	optimization of the production yield	Development and evaluation of a general framework for integrated control based on GNN and MARL	Multi-Agent Reinforcement Learning (MARL)	private simulated data
Deep multi-agent reinforcement learning for multi-level preventive maintenance in manufacturing systems [75]	Maintence Strategies and Quality	Multi-level preventive manufacturing (PM) scheduling	Implementation of a MARL algorithm to obtain PM decision making process policies in serial production lines	Multi-Agent Reinforcement Learning (MARL)	private simulated data
Deep reinforcement learning based scheduling within production plan in semiconductor fabrication [40]	Scheduling	Production planning and scheduling	A DRL based scheduling method is proposed to fulfil the production plan for semiconductor fabrication	Deep Q Network (DQN)	private simulated data
Dynamic Control of a Fiber Manufacturing Process Using Deep Reinforcement Learning [32]	Process control	Fiber drawing system	introduction of a compact fiber drawing system and development of a DRL-based strategy for diameter tracking	model-free deep reinforcement learning (DRL)	private simulated data
Reinforcement learning for online optimization of job-shop scheduling in a smart manufacturing factory [100]	Scheduling	job-shop scheduling problem (JSSP) in dynamic systems	This paper presents a smart scheduler for online scheduling low-volume-high-mix orders in a smart manufacturing factory	Deep Q Network (DQN) with composite rewards	private simulated data
Reinforcement learning approach to scheduling of precast concrete production [33]	Scheduling	scheduling problems of precast concrete PC) production	This study proposed a DQN approach to solve the PC scheduling problem of minimizing total tardiness.	Deep Q Network (DQN) with PC productio n	private simulated data

				scheduling simulator (PC-DQN).	
Task Allocation in Human–Machine Manufacturing Systems Using Deep Reinforcement Learning [30]	Scheduling	Task allocation (dynamic scheduling) problem in human–machine manufacturing systems	To improve task allocation in human-machine manufacturing system a NN-RL algorithm, which considers fatigue accumulation and task competence level of human operators, while achieving faster mean flowtime compared to classical dispatching rules is proposed	Neural network (NN)-based supervised learning	private simulated data
Optimisation of manufacturing process parameters for variable component geometries using reinforcement learning [103]	Autonomous Manufacturing	optimal manufacturing parameters problem	RL based approach for estimation of optimal manufacturing parameters for variable component geometries. The presented approach trains a function P which takes the component geometry as input and directly estimates optimal process parameters (output).	RL	private simulated data
Explainable Deep Reinforcement Learning For Production Control of job shop manufacturing system [36]	Process control	production planning and control (PPC) problem	a multi-Agent system (MAS) based on DRL is developed to realize short reaction time and high decision quality in a manufacturing user-centric systems	Multi Agent Reinforcement Learning (MARL) algorithm	private simulated data

A Dynamic Chemical Production Scheduling Method based on Reinforcement Learning [95]	Scheduling	Dynamic chemical production scheduling problem	The paper adopts an algorithm based on the PPO to add short-term state inventory to improve the state function, so that the policy network can achieve more accurate scheduling according to the urgency of orders and enhance the stability of the algorithm.	Proximal Policy Optimization (PPO) algorithm based on the Advantage Actor-Critic (A2C) framework	private simulated data
Reinforcement learning and digital twin-based real-time scheduling method in intelligent manufacturing systems [96]	Scheduling	Optimization efficiency and decision-making responsiveness problem in intelligent manufacturing	Development of a RL and digital twin-based real-time scheduling method of Automated guided vehicle (AGVs), called twins learning, to satisfy multiple objectives simultaneously	deep Q-Learning network algorithm	private simulated data
Discovery of customized dispatching rule for single-machine production scheduling using deep reinforcement learning [10]	Scheduling	production scheduling-dispatching rule problem	Using parameters obtained readily within the digital twin setting, this paper investigates the application of deep reinforcement learning to select customized dispatching rules formed by weighted combinations of production parameters on a single machine production scheduling problem.	deep Q-Learning Markov decision process (MDP) algorithm	private simulated data
A Novel Reinforcement Learning-based Unsupervised Fault Detection for Industrial Manufacturing Systems [1]	Maintenance Strategies and Quality	fault detection (FD) systems problem	Development of a RL algorithm (DDQN with prioritized experience replay) to optimize fault detection system. The paper also validates the effectiveness of the	double deep-Q network (DDQN) with prioritized experienc	private simulated data

			proposed algorithm on real steel plant data	e replay (PER)	
An Adaptive Reinforcement Learning-Based Scheduling Approach with Combination Rules for Mixed-Line Job Shop Production [102]	Scheduling	Flexible job shop scheduling problem (FJSP).	This paper presents an adaptive scheduling method for mixed-line job shop scheduling with combined processing constraints. It is also validated with experiments in a smart manufacturing setting (mixed-line job shop of missile structural parts in Shanghai).	RL algorithm using a LinUCB-based scheduling method	private simulated data
Collaborative Clustering Parallel Reinforcement Learning for Edge-Cloud Digital Twins Manufacturing System [16]	Autonomous Manufacturing	job shop scheduling (JSS) problem in an Edge-Cloud Digital Twins Manufacturing System	Construction of a novel edge-cloud collaborative architecture to support the DT-based job shop scheduling (JSS) application and optimize the running position of DT-based applications to minimize the total delay according to the application attributes and cloud-edge resources status.	collaborative clustering parallel Q-learning (CCPQL) and prediction-based CCPQL algorithm	private simulated data
Deep Reinforcement Learning-Based Job Shop Scheduling of Smart Manufacturing [101]	Scheduling	Job-Shop Scheduling Problem (JSSP) in Smart Manufacturing	The paper proposes a problem formulation for JSSP as a sequential decision-making problem, designs the model to represent the scheduling policy based on Graph Isomorphism Network, then it introduces the training algorithm as the actor-critic network algorithm.	Deep Reinforcement Learning with an Actor-Critic algorithm (DRLAC).	private simulated data
Reinforcement Learning based on Stochastic Dynamic Programming	Maintenance Strategies and Quality	maintenance actions planning	In this paper, a stochastic dynamic programming model is	Q-learning algorithm with a	private simulated data

for Condition-based Maintenance of Deteriorating Production Processes [20]			developed for maintenance planning on a deteriorating multistate production system	partial observable Markov decision process (POMDP)	
Reconfigurable manufacturing system scheduling: a deep reinforcement learning approach [78]	Scheduling	scheduling problem of Reconfigurable Manufacturing Systems (RMS) on multiple products	Development of a DDQN-based scheduling policy training approach in Reconfigurable Manufacturing Systems (RMS)	Double DQN-based algorithm with Prioritised Experience Replay (PER)	private simulated data
Predictive Maintenance Decision Making Based on Reinforcement Learning in Multistage Production Systems [51]	Maintenance Strategies and Quality	Predictive maintenance problem in multistage production systems	A reinforcement learning approach is proposed to optimize the production and maintenance cost.	RL	private simulated data
A flexible manufacturing assembly system with deep reinforcement learning [44]	Motion Planning	improving the flexibility of the assembly process	To improve the flexibility of assembly lines, the article proposes a deep reinforcement learning and digital twin-based approach which focus on motion planning, precision, and safety of the manufacturing system	Deep Deterministic Policy Gradient (DDPG) based assembly algorithm	private simulated and real data
Multi-objective optimization of the textile manufacturing process using deep-Q-network based multi-agent reinforcement learning [22]	Process control	textile process optimization problem in a multi-agent system	Development of a DQN based MARL system to solve optimization problems and its validation in the textile ozonation process and enzyme washing process	self-adaptive DQN (Deep-Q-Network) - based multi-agent reinforcement learning (MARL)	private simulated data

Modular production control using deep reinforcement learning: proximal policy optimization [54]	Process control	Modular production control problem in the automotive industry	Modular production systems are a new field in the automotive industry and the article proposes a Proximal Policy Optimization DRL algorithm to address modular production control	Proximal Policy Optimization (PPO) DRL method	private simulated data
A fuzzy hierarchical reinforcement learning based scheduling method for semiconductor wafer manufacturing systems [82]	Scheduling	Production scheduling in semiconductor wafer manufacturing system (SWFS) problem	This paper proposed a fuzzy hierarchical reinforcement learning (FHRL) approach to control cycle time (CT) in the scheduling of a SWFS, which is a typical complex large-scale manufacturing system.	fuzzy hierarchical reinforcement learning (FHRL)	private simulated data
Towards Self-X cognitive manufacturing network: An industrial knowledge graph-based multi-agent reinforcement learning approach [99]	Autonomous Manufacturing	Self-X (e.g. self-configure, self-optimize, and self-adjust/adaptive/healing) cognitive manufacturing network efficient management problem	This research introduces an IKG-based MARL approach for automatic manufacturing task fulfilment with self-configuration and self-optimization capabilities, towards the proposed Self-X cognitive manufacturing network.	industrial knowledge graph (IKG)-based multi-agent reinforcement learning (MARL)	private simulated data
Joint optimization of preventive maintenance and production scheduling for multi-state production systems based on reinforcement learning [92]	Maintenance Strategies and Quality	Preventive maintenance and production scheduling problems	The paper investigates the integrated optimization of preventive maintenance and production scheduling for multi-state single-machine production systems with the deterioration effect, therefore a novel HR learning algorithm was presented to tackle	heuristic reinforcement (HR) learning algorithm	private simulated data

			MDP model based on R-learning.		
Reinforcement Learning for Statistical Process Control in Manufacturing [80]	Process control	manufacturing optimization (cost reduction while considering the rate of good products)	The paper introduced the concept and the solution to place Reinforcement Learning (RL) into Statistical Process Control (SPC) in manufacturing. The formulated manufacturing goal was to minimize the production unit cost while keeping the ratio of good products on a high level and it was developed and validated using a TD learning algorithm.	Temporal Difference (TD) learning algoritm	private simulated and real data
Dynamic matching with deep reinforcement learning for a two-sided Manufacturing-as-a-Service (MaaS) marketplace [61]	Scheduling	Dynamic matching with deep reinforcement learning for a two-sided Manufacturing-as-a-Service (MaaS) marketplace	real-time decision making for suppliers participating in a manufacturing-as-a-service (MaaS) marketplace	Deep-Q-Network (DQN)	private simulated data
Fault-Tolerant Control of Programmable Logic Controller- Based Production Systems With Deep Reinforcement Learning [104]	Autonomous Manufacturing	system availability in logic controller based automated production system	The authors explicitly focused on automated production system (aPS). To overcome the challenges of an exploding action space and a missing global coordinate system for the tracking of workpieces, a hierarchical MAS with a separate coordinate predictor per agent was suggested and validated.	hierarchical multi-agent deep reinforcement learning approach	private simulated data
Digital Twin and Reinforcement Learning-Based Resilient Production Control for	Process control	efficient personalized production	To improve the cyber-physical production systems (CPPS) for enhancing the process	Q-learning algorithm	private simulated data

Micro Smart Factory [64]			and systematic efficiency of micro smart factory (MSF), the DT and RL-based resilient production control methods are proposed in this paper.		
Designing an adaptive production control system using reinforcement [35]	Scheduling	adaptive order dispatching optimizing	This paper addresses the design of RL to create an adaptive production control system by the real-world example of order dispatching in a complex job shop.	RL-algorithm with fixed state informatio n	private simulated and real data
Deep Reinforcement Learning-based maintenance decision-making for a steel production line [81]	Maintenance Strategies and Quality	maintenance optimization	This work proposes a DRL policy for a scrap-based steel production line where maintenance decisions are taken in real-time by the monitoring condition of the production line aiming to minimize the long-run maintenance cost per unit of time.	Double Deep Q Network (DDQN)	private simulated data
Control of Shared Production Buffers: A Reinforcement Learning Approach [56]	Scheduling	buffer control problem	This paper proposes Q-learning algorithm for buffer control problem for stochastic flow lines with shared production buffers.	Q-learning algorithm	private simulated data
Demonstrating Reinforcement Learning for Maintenance Scheduling in a Production Environment [27]	Maintenance Strategies and Quality	maintenance scheduling problem	In this paper the usability of RL, notably Q-learning, for finding an optimal strategy to schedule maintenance capacity in a realistic production environment has been demonstrated.	Q-learning algorithm	private simulated data
A reinforcement learning model for material handling task	Scheduling	material handling problem	This study analyzes the application of RL for material handling tasks	Q-learning algorithm	private simulated data

assignment and route planning in dynamic production logistics environment [29]			in Smart Production Logistics (SPL) in the automotive industry. In particular, this study addressed the routing of Automated Guided Vehicles (AGVs) for material handling including dynamic aspects.		
Integrated Planning and Scheduling for Customized Production using Digital Twins and Reinforcement Learning [55]	Scheduling	planning and scheduling problem for dynamic/customised production	In this paper, it is presented a digital twin based self-learning process planning approach using Deep-Q-Network that can identify optimized process plans and workflows for the simultaneous production of personalized products.	Deep-Q-Network (DQN)	private simulated data
Modelling Production Scheduling Problems as Reinforcement Learning Environments based on Discrete-Event Simulation and OpenAI Gym [39]	Scheduling	Production scheduling problem	The paper presented a method that guides the modelling of production scheduling problems as RL environments. It involves the application of Discrete Event Simulation (DES) and the OpenAI Gym interface.	Discrete Event Simulation (DES) based algorithm	private simulated data
A Deep Reinforcement Learning approach for the throughput control of a FlowShop production system [53]	Process control	throughput control of a Flow-Shop production system problem	To achieve a throughput target, a Deep Q-Network (DQN) is developed and used to define the constant WIP quantity in the system.	Deep-Q-Network (DQN)	private simulated data
Simultaneous Production and AGV Scheduling using Multi-Agent Deep	Scheduling	Flexible Job Shop Scheduling Problem (FJSSP), including the coordination of the	In this paper, a concept for simultaneous machine job scheduling with transport planning in a flexible job shop	Multi Agent Reinforcement Learning	private simulated data

Reinforcement Learning [68]		Automated Guided Vehicles (AGVs)	using a Multi Agent Reinforcement Learning (MARL) algorithm was presented.	(MARL) algorithm	
Reinforcement Learning Based Production Control of Semi-automated Manufacturing Systems [60]	Process control	production control problem	In this work, Digital Twin (DT) and RL are combined to derive a production control logic in a semi-automated production system based on the chaku-chaku principle.	proximal policy optimizati on algorithm (PPO)	private simulated and real data
A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence [90]	Scheduling	adaptive manufacturing strategies problem	In this work, a novel approach is proposed to utilize digital twin simulation and communication technologies to create virtual counterparts of robot manufacturing systems, on which the intelligent scheduler based on Deep Reinforcement Learning can be safely trained to optimize smart manufacturing task.	Deep-Q-Network (DQN) and Double Deep Q Learning (DDQN)	private simulated and real data
A Deep Reinforcement Learning Based Scheduling Policy for Reconfigurable Manufacturing Systems [77]	Scheduling	scheduling policy for Reconfigurable manufacturing systems (RMS)	This paper mainly focuses on optimising RMS scheduling using Deep Q Learning (DQL) by reducing reconfiguring actions and while minimising the makespan.	Discrete-event simulation (DES) based Deep-Q-Network (DQN) and Double Deep Q Network (DDQN)alg orithm	private simulated data
Reinforcement Learning With Composite Rewards for Production	Scheduling	real time production scheduling problem	This paper presents an AI scheduler for online and dynamic	Q-learning algorithm	private simulated data

Scheduling in a Smart Factory [76]			scheduling of manufacturing jobs in a smart factory. The RL method equips the proposed system with self-organizing and self-learning capabilities under uncertainty		
Two-time scale reinforcement learning and applications to production planning [97]	Scheduling	optimal control problems of dynamic systems	This paper is focused on two-time-scale RL. A production planning system is used throughout as an example to demonstrate ideas and preliminary results and Monte Carlo simulations are used as 'data' provider for training and validation.	two-time-scale RL.	private simulated data
Deep reinforcement learning based preventive maintenance policy for serial production lines [24]	Maintenance Strategies and Quality	preventive maintenance (PM) problem	The PM decision making in a serial production line is a complex problem due to its exploding state space and complicated interactions among machines. The problem is proposed to be solved using a DRL approach in this paper.	Double Deep Q Network (DDQN)	private simulated data
Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system [34]	Scheduling	production planning and job scheduling problem in smart manufacturing	In this paper, it is presented a smart manufacturing system using a multiagent system and RL, which is characterized by machines with intelligent agents to enable a system to have autonomy of decision making, sociability to interact with other systems, and intelligence to	Deep Q Network (DQN)	private simulated data

			learn dynamically changing environments.		
Reinforcement learning for combined production-maintenance and quality control of a manufacturing system with deterioration failures [62]	Maintence Strategies and Quality	optimal joint production, maintenance, and product quality control policies	This research paper examined a stochastic system that is experiencing frequent degrading failures and addressed the problem of finding optimal joint control policies in respect to an objective function that maximizes the total profit of the described system.	Q-learning algorithm	private simulated data
A Reinforcement Learning Approach to Robust Scheduling of Semiconductor Manufacturing Facilities [26]	Scheduling	scheduling problem in semiconductors' industry	In this article, it is presented a scheduling method for minimizing the make span of semiconductor manufacturing systems through the Q-learning based on a neural network (NN).	Q-learning algorithm	private simulated data
Reinforcement learning for facilitating human-robot-interaction in manufacturing [59]	Autonomous Manufacturing	optimization of human-robot-interaction in manufacturing	The work presented is intended to illustrate the applicability of reinforcement learning to the problem of robotic control within manufacturing,specifically in cases where there is significant variation introduced by human operators.	Deep Q Network (DDQN)	private simulated data
Intelligent scheduling of discrete automated production line via deep reinforcement learning [74]	Scheduling	scheduling problem in single product discrete automated production line	This paper proposes a deep RL-based online scheduling method for discrete automated production line. A Discrete Event Simulation (DES) environment is built to provide an intelligent	Discrete-event simulation (DES) based RL algorithm	private simulated data

			and efficient environment for RL model, reaching a competitive performance of online intelligent scheduling policy		
Reinforcement learning for an intelligent and autonomous production control of complex job-shops under time constraints [5]	Scheduling	order dispatching in a complex environment including time constraints	In this paper a Q-learning algorithm is applied in combination with a process-based discrete-event simulation to train a self-learning, intelligent, and autonomous agent for the decision problem of order dispatching in a complex job shop with strict time constraints.	Deep Q Network (DQN)	private simulated data
Model-free Adaptive Optimal Control of Episodic Fixed-horizon Manufacturing Processes Using Reinforcement Learning [15]	Process control	adaptive optimal control of episodic fixed-horizon manufacturing processes with varying process conditions problem	A Q-learning-based method for adaptive optimal control of partially observable episodic fixed-horizon manufacturing processes is developed and studied. The resulting algorithm is instantiated and evaluated by applying it to a simulated stochastic optimal control problem in metal sheet deep drawing.	fixed horizon manufacturing processes (FHMP)-Q-Control algorithm	private simulated data
A Model-Based Reinforcement Learning and Correction Framework for Process Control of Robotic Wire Arc Additive Manufacturing [2]	Process control	process study and control of Multi-Layer Multi-Bead (MLMB) deposition in Robotic Wire Arc Additive Manufacturing (WAAM)	This paper presents an integrated model-based RL-correction framework for in-situ MLMB process learning of robotic WAAM, as well as the preliminary experimental study of the learning framework	model-based parallel reinforcement learning	private simulated and real data

			on a physical robotic WAAM system performing printing tasks for two different materials.		
Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network [23]	Scheduling	dynamic scheduling problem of flexible manufacturing systems (FMSs)	To solve the dynamic scheduling problem of an FMSs involving shared resources, route flexibility, and stochastic arrivals of raw products, this paper proposed a novel Petri-net-based dynamic scheduling approach via DQN with graph convolutional network (GCN).	Deep Q Network (DQN)	private simulated data
Logistics-involved QoS-aware service composition in cloud manufacturing with deep reinforcement learning [46]	Scheduling	Cloud manufacturing service composition (CMfg-SC) problem.	A DRL algorithm, named PD-DQN, which combines the basic DQN algorithm, the dueling architecture, and the prioritized replay mechanism was used for CMfg-SC	A dueling Deep Q-Network (DQN) with prioritized replay named PD-DQN	private simulated data
Deep Reinforcement Learning for Semiconductor Production Scheduling [7]	Scheduling	Semiconductor production scheduling problem	In this paper DRL is applied to production scheduling in semiconductor complex job shops utilizing cooperative Deep Q Network (DQN) agents. The DQN agents, which use deep neural networks for decision making, are trained in a RL environment with user-defined flexible objectives to optimize production scheduling.	Deep Q Network (DQN)	private simulated data

Optimization of global production scheduling with deep reinforcement learning [87]	Scheduling	production scheduling problem	In an RL environment cooperative DQN agents, which utilize deep neural networks, are trained with user-defined objectives to optimize scheduling..	Deep Q Network (DQN)	private simulated data
Reinforcement Learning-Based and Parametric Production-Maintenance Control Policies for a Deteriorating Manufacturing System [3]	Maintence Strategies and Quality	joint production/maintenance control policies problem	In this paper the problem of integrated production/maintenance control for a deteriorating, stochastic production/inventory system was investigated. A novel approach, based on RL, for deriving optimal or near-optimal policies was proposed	Q-learning algorithm	private simulated data
Optimized Adaptive Scheduling of a Manufacturing Process System with Multi-Skill Workforce and Multiple Machine Types: An Ontology-Based, Multi-Agent Reinforcement Learning Approach [70]	Scheduling	optimized manufacturing scheduling problem	This research develops a multi-agent reinforcement learning approach for the optimal scheduling of a manufacturing system of multi-stage processes for multiple types of products with various machines and a multi-skilled workforce.	Multi-agent approximate Q-learning	private simulated data

Table 1. Selected papers

7. Applications in manufacturing

To better understand the manufacturing applications of RL an analysis considering the different area of implementation of the different algorithms used in the papers in table 1 must be conducted.

I've then decided to structure the documents' review considering the listed area of application:

1. Robotics for motion planning

2. Scheduling
3. Process Control
4. Autonomous Manufacturing
5. Maintenance Strategies and Quality
6. Sustainable Manufacturing

7.1.1. Applications of Reinforcement Learning in Industrial Robotics for motion planning

- User-guided motion planning with reinforcement learning for human-robot collaboration in smart manufacturing [79]

The paper introduces a user-guided motion planning algorithm coupled with reinforcement learning (RL) to empower robots to autonomously generate motion plans for new tasks by learning from a small number of kinesthetic human demonstrations. To achieve adaptive motion planning in the face of task changes or new requirements, a movement library is created. The features embedded in the library are then mapped to specific task segments based on the trained motion planning policy using Q-learning. A new task can be learned as a combination of features in the library or, if the library is insufficient, further human demonstrations may be required. The trained motion planning policy's performance is evaluated in an assembly and loading/unloading scenario for three new tasks: a transferring task (where the end-effector transfers a cup of water while avoiding an obstacle), a filling-and-pouring task, and an assembling task. Each of these tasks is successfully executed 20 times in the trials.

- Spatiotemporal path tracking via deep reinforcement learning of robot for manufacturing internal logistics [17]

A method for controlling the time-space path tracking of robots in weaving scenes, facilitating intelligent logistics and storage, is introduced. The primary contributions are outlined as follows: The proposal of a hybrid Deep Reinforcement Learning (DRL) framework for path tracking is a key aspect. This framework integrates scene feature models, addressing the challenge of prolonged training phases in DRL, more suitable for simulated environments or adversarial games than real scenarios. The integration of scene features into partially observable Markov decision models is emphasized, particularly through the introduction of a Dynamic Observing Markov Decision Process (DOMDP). An empirical optimization method, centered around observation difference ranking, is introduced to enhance stability in convergence results, considering data heterogeneity and external influences. For assessing performance, PyBullet is employed as the physical engine for scenario simulation, and TensorFlow is utilized to implement the network architecture. The proposed learning framework, dynamic observation deterministic strategy gradient (DODPG), is specifically designed and evaluated against other algorithms, namely twin-delayed deep deterministic policy gradient (TD3) and Deep Deterministic Policy Gradient (DDPG). Across all simulated scenarios, the DOMDP algorithm consistently achieves optimal results, surpassing the performance of DDPG and TD3. Notably, DODPG demonstrates effective control, maintaining the average instantaneous tracking distance error of the robot generally within 0.005 m.

- Reinforcement Learning Enabled Self-Homing of Industrial Robotic Manipulators in Manufacturing [31]

This paper presents a non-vision, model-free, off-policy reinforcement learning-based approach, specifically Soft Actor-Critic (SAC), designed for enabling self-homing capability in industrial robotic cells. The primary focus is on the homing task, where a robotic arm returns to its initial/home position from any location in a robotic cell without collisions, robot singularities, and within joint limits. The proposed approach eliminates the need for manual programming of robot manipulators. The approach is characterized by being model-free and utilizing off-policy reinforcement learning. It adopts a parallel agent setting, where multiple agents learn simultaneously to enhance exploration and learning. Training is conducted in a simulation environment generated by the mechanical design of an actual robotic cell. The agents are assumed to sense the unknown robotic cell environment, which is pre-encoded in the state definition. The paper investigates the impact of curriculum on the agent's learning, comparing two curriculum choices with a non-curriculum baseline. The training setting involves a parallel-agent, multi-process training approach to improve exploration in the state space, with experiences shared among agents via shared memory. Trained models are subsequently deployed in real robotic manufacturing cells for brazing and assembly applications in aircraft engines. The success rate of the Deep Reinforcement Learning (DRL) approach is reported as 98%, surpassing joint motion (57%), linear motion (29%), and human-generated (8%) methods in homing the robot to the home position. This research highlights the effectiveness of using reinforcement learning, particularly SAC, for automating the homing task in industrial robotic cells, with successful real-world deployment and performance evaluations.

- A flexible manufacturing assembly system with deep reinforcement learning [44]

This article introduces a comprehensive solution encompassing the automated planning of assembly motions and a monitoring system for production lines. In the planning phase, a digital twin model of the assembly line is constructed, followed by the training of a deep reinforcement learning agent to carry out the assembly of workpieces. In the production phase, the digital twin model is utilized to monitor the assembly lines and predict potential failures.To validate the effectiveness of the proposed system, a peg-in-hole assembly experiment was conducted, resulting in an impressive 90% success rate for a single assembly attempt. Notably, no collisions occurred in the real-world scenario throughout the entire experiment. This highlights the robustness and reliability of the proposed solution in achieving successful and collision-free assembly processes.

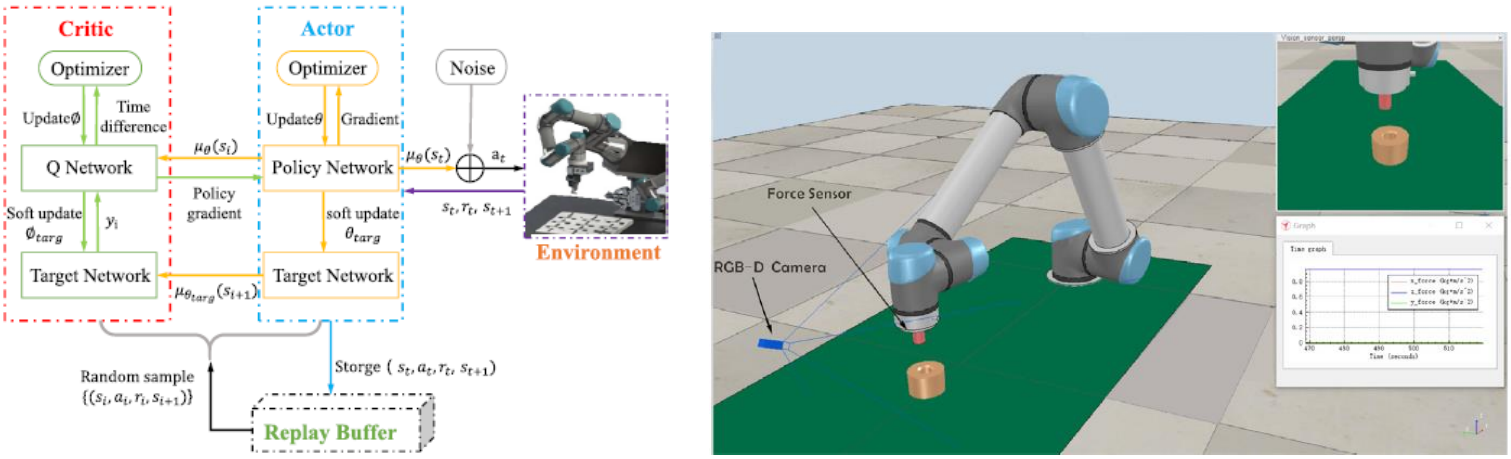

Figure 7. Overall training procedure of reinforcement learning agent and Digital twin model of the peg-in-hole assembly workspace [44]

7.1.2. Pioneering Contributions to the State of the Art:

These pioneering articles have marked substantial contributions to the realm of Reinforcement Learning (RL) in the context of robotics motion planning applications. Their novel methodologies showcase advancements in addressing challenges related to complex and dynamic environments, high-dimensional state spaces, sparse rewards, and non-linear and non-convex optimization problems within the field of robotics. Importantly, these proposed methods have been validated through rigorous experimentation on actual robot platforms, affirming their practical efficacy and real-world applicability.

7.1.3. Advantages of Reinforcement Learning in Robotics for motion planning:

Reinforcement Learning (RL) algorithms have emerged as a powerful tool in enabling robots to learn from their environment and make informed decisions based on received feedback. Particularly in the domain of robotics, RL has been increasingly applied to motion planning, a crucial aspect involving the determination of a sequence of actions for a robot to navigate from one location to another. In comparison to conventional motion planning methods, RL offers several distinct advantages.

Primarily, RL algorithms exhibit a remarkable capacity to handle complex and dynamic environments. Unlike traditional motion planning approaches that rely on predetermined rules and assumptions about the environment, RL algorithms demonstrate adaptability to environments characterized by complexity and frequent changes. This adaptability stems from their ability to learn from the environment, allowing robots to navigate effectively in dynamic settings. Another noteworthy advantage of RL algorithms is their proficiency in managing high-dimensional state spaces. Traditional motion planning methods may encounter challenges in coping with state spaces characterized by an extensive range of possible states. RL algorithms, however, overcome this hurdle through the utilization of function approximators, such as neural networks, enabling them to navigate efficiently

within high-dimensional state spaces. Sparse rewards pose a common challenge in motion planning applications, where robots may not receive feedback for every action taken. RL algorithms stand out in handling sparse rewards, leveraging techniques such as Q-learning and policy gradients to learn effectively from limited feedback. This adaptability ensures that robots can make informed decisions even in scenarios with sparse reward signals, a feat that traditional methods may find challenging.

Furthermore, RL algorithms exhibit prowess in addressing non-linear and non-convex optimization problems, prevalent in motion planning applications. Traditional methods often struggle with such complexities, but RL algorithms, particularly those employing deep reinforcement learning techniques, showcase an ability to navigate non-linear and non-convex optimization challenges effectively.

7.1.4. Areas for improvement and future directions:

Considering the papers that focused on the application of RL in Robotics for motion planning the principal areas for improvement are:

- improvements in RL-based motion planning algorithms should focus on effectively learning from a wide range of human demonstrations.
- addressing challenges in dynamically sequencing tasks in response to real-time changes in manufacturing environments enhances the efficiency and adaptability of robots in complex processes.
- enhancements in RL algorithms for self-homing should focus on addressing challenges related to accuracy and robustness.
- Improving the adaptability of scheduling decisions to dynamic changes in the assembly environment, ensuring efficient utilization of robotic resources.
- addressing challenges related to real-time adaptability, safety, and responsiveness enhances the quality of collaboration between humans and robots in manufacturing settings.
- optimizing the control of robotic deposition processes, ensuring high-quality and precise manufacturing outcomes.

Reinforcement learning (RL) has demonstrated significant potential in addressing robotics' motion planning in manufacturing. Nevertheless, several challenges and constraints must be overcome to advance future applications. The forthcoming directions are:

1. Efficient Learning from Limited Demonstrations: RL algorithms typically demand substantial training data for optimal performance, a process that can be resource intensive. Future research should strive to devise RL algorithms capable of efficient learning from fewer demonstrations, reducing the time and cost associated with training. The research 'User-guided motion planning with reinforcement learning for human-robot collaboration in smart manufacturing [79]' which proposes a RL method to enable robots to automatically generate their motion plans for new tasks by learning from a few kinaesthetic human demonstrations. This approach, grounded in human demonstrations, attains scheduling decisions near optimality with minimized training data.

2. Robustness to Environmental Variations: RL algorithms may exhibit sensitivity to environmental changes, resulting in suboptimal performance. To counteract this challenge, forthcoming research should concentrate on enhancing the robustness of RL algorithms to environmental variations, encompassing factors such as changes in lighting, temperature, or humidity. The work 'Spatiotemporal path tracking via deep reinforcement learning of robot for manufacturing internal logistics [17] can be a starting point as the proposed method can learn the optimal control policy from raw sensor data and achieve high-precision path tracking in complex environments.

3. Adaptive Learning for Dynamic Environments: Manufacturing environments are inherently dynamic, featuring fluctuations in production schedules, equipment configurations, and personnel assignments. Future research endeavours should aim to formulate RL algorithms equipped to adapt to these dynamic conditions, facilitating real-time learning for more efficient and effective scheduling.

4. Integration with Complementary Technologies: RL algorithms possess the potential for integration with various technologies, including computer vision, natural language processing, and machine learning. Exploring these integrations and developing novel applications that leverage the strengths of multiple technologies should be a focal point for future research in enhancing RL applications in robotics. 'A flexible manufacturing assembly system with deep reinforcement learning [44]' proposes a digital twin enhanced assembly method with deep reinforcement learning. The proposed method can be a starting point as it can enable the robot to learn the optimal assembly policy from raw sensor data and achieve high-precision assembly in a flexible manufacturing environment.

7.2. Applications of Reinforcement Learning in Scheduling

- Simulation and deep reinforcement learning for adaptive dispatching in semiconductor manufacturing systems [72]

A dispatching and resource allocation approach for a semiconductor manufacturing system is developed, leveraging a Deep Q-Network (DQN). The system is modeled and simulated using a data-driven agent-based Discrete Event Simulation (DES) in Arena simulation. This simulation environment serves as the training ground for the DQN agents, responsible for dispatching products and allocating equipment during the simulation execution. Each agent oversees a single station with one or more equipment units. When a dispatching action is required, the simulation furnishes the agents with the current state of the system. The agent, based on this state, selects an appropriate action, which is then implemented by the system. The agent receives a reward corresponding to the performance of the station and the overall system. The DQN approach is contrasted with the currently employed heuristics-based dispatching, specifically the First-In-First-Out (FIFO) approach. Through the simulation, DQN agents learn to enhance the performance of stations and the entire system, demonstrating the ability to collaborate, especially in challenging scenarios such as timers' processes. Results indicate an improvement in system performance with DQN agents, manifesting as a general increase in throughput, reduction in the total non-value-added time percentage, and decreased instances of timer exceeding and resulting waste.

- Sequence generation for multi-task scheduling in cloud manufacturing with deep reinforcement learning [66]

The paper addresses the challenge of multi-task scheduling in cloud manufacturing and introduces a corresponding approach based on Deep Reinforcement Learning (DRL), incorporating sequence generation techniques. Initially, two sequence generation algorithms are proposed, tasked with generating scheduling sequences for multiple composite tasks by ranking tasks and subtasks. Subsequently, two scheduling approaches, leveraging Deep Q-Network (DQN) and Double DQN, are introduced in conjunction with the sequence generation techniques. The performance of these algorithms is evaluated against seven baseline approaches, including random, round-robin, earliest, sensible, minimum execution time (min-t), minimum machining cost (min-c), and maximum reliability (max-r) scheduling algorithms. The evaluation metrics encompass makespan, total cost, and average reliability. Results indicate the superiority of the DQN-based approach over all baseline methods, with Double DQN demonstrating significant advantages over DQN.

- Scheduling of decentralized robot services in cloud manufacturing with deep reinforcement learning [49]

The paper addresses the optimization of robot services in cloud manufacturing through the introduction of a novel scheduling model based on Deep Reinforcement Learning (DRL), incorporating both Deep Q-Network (DQN) and Double DQN. The model considers the quality and performance of both robot and logistics services, aiming to maximize service quality while minimizing service cost. Comparative evaluations were conducted with DQN, Double DQN, and three benchmark scheduling approaches: random scheduling, round-robin scheduling, and earliest scheduling. Using Python-based simulation programs, the results indicate that the DDQN-based scheduling approach outperforms all other methods across various indices. Furthermore, a multiple linear regression analysis was employed to assess the influencing degrees of different indicators, revealing that logistics service reliability and execution times are the most influential factors on overall service quality.

- Reinforcement learning-based dynamic production-logistics-integrated tasks allocation in smart factories [41]

The paper focuses on harnessing the intelligence of smart connected resources to autonomously allocate production and logistics tasks in smart factories (SFs). The system is conceptualized as an autonomous decision-making manufacturing system with Industrial Internet of Things (IIoT) support, aiming to coordinate and synchronize the allocation of manufacturing tasks through resource bidding in SFs. A dynamic production-logistics-integrated tasks allocation model is proposed, considering the orders makespan and resource utilization as the objective function. Both production and logistics resources autonomously communicate and interact to bid for dynamic production-logistics integrated operations. The study employs a reinforcement learning (RL) algorithm, specifically the Q-learning algorithm, to make operational decisions for each job step based on in-situ data during the manufacturing process. A demonstrative case illustrates that the RL-based model outperforms centralized scheduling systems, particularly in handling production-logistics-integrated tasks allocation problems in SFs characterized by dynamic and small-batch individualized orders.

- Reinforcement learning based trustworthy recommendation model for digital twin-driven decision-support in manufacturing systems [67]

This paper introduces an innovative digital twin decision support framework that integrates recommendation systems with reinforcement learning (RL) algorithms, trust and similarity measures. The aim is to enhance the accuracy and reliability of recommendations, leading to improved efficiency, reduced downtime, and increased production output for optimized manufacturing processes. The SimQL model, comprising a trust model, a Q-learning-based RL algorithm, and similarity measures, is formally specified to address common challenges associated with recommendation systems. These challenges include user trust, decision-making time, cold-start, and data sparsity issues. The proposed model is experimentally validated in a manufacturing case study involving a battery pack assembly line. Comparative analysis with state-of-the-art recommendation models demonstrates the effectiveness of the SimQL model, showcasing superior accuracy.

- Multi-agent deep reinforcement learning for task offloading in group distributed manufacturing systems [91]

To address challenges in the task offloading process in cloud manufacturing, a mixed-integer programming model has been developed to reduce task calculation latency. The problem is divided into two sub-problems: 1) Defining priorities for tasks in near real-time. 2) Determining if the task should be offloaded to the cloud. A multi-agent deep reinforcement learning framework with an attention mechanism (MaDRLAM) is proposed to tackle these sub-problems. The MaDRLAM framework involves two agents, each handling one sub-problem. Each agent is composed of an encoder and a decoder, based on the Transformer structure with added Pointer networks to address the proposed decision problem.The novel aspect of the MaDRLAM framework lies in its attention mechanism and Transformer structure. Additionally, an improved multi-actor and single-critic strategy based on the REINFORCE algorithm is designed to train the proposed MaDRLAM. Computational experiments are conducted on instances with varying numbers of tasks, different task data sizes, and diverse cloud computing capacities. The results demonstrate that the proposed framework efficiently finds solutions with a GAP value of less than 1% within 1 second for each instance. The framework proves competitive in both solution accuracy and solution time when compared with other offloading strategies.

- Logistics-involved task scheduling in cloud manufacturing with offline DRL [84]

This paper introduces an offline DRL scheduling algorithm designed for addressing CMfg-SPs. Unlike many existing DRL-based methods that train models online, the proposed method conducts offline training using historical data. This departure from online training mitigates the risks associated with using potentially unstable DRL models for generating scheduling schemes and enhances the utilization of historical data and large deep learning models. The research represents an early exploration of applying offline DRL and DT architecture to solve CMfg-SPs, aiming to retain the benefits of online DRL while minimizing the risk of online trial-and-error. The approach's applicability extends beyond CMfg-SPs to other scheduling problems. Experimental results, using a case study of the automobile structure part scheduling problem, affirm the effectiveness of the proposed method, with sensitivity analysis offering insights for adjusting hyperparameters.

- Inverse Reinforcement Learning Framework for Transferring Task Sequencing Policies from Humans to Robots in Manufacturing Applications [58]

To address the growing demand for skilled individuals in complex production processes, manufacturers are increasingly turning to the deployment of robots. This study introduces an inverse reinforcement learning approach to solve the challenge of task sequencing for robots engaged in intricate manufacturing processes. The success of the manufacturing process heavily depends on the sequence in which subtasks are executed. Therefore, the focus of this work is on modeling the expert's policy for sequencing subtasks to attain desirable outcomes in the process. The robots are trained using expert demonstrations, which are collected in a dataset and utilized to construct the sequence policy. By learning weights that prioritize the expert's sequence, the method successfully achieves the lowest cost for all demonstration tools in both real-world and synthetic data scenarios.

- Graph neural networks-based scheduler for production planning problems using reinforcement learning [21]

This paper introduces a novel framework called GraSP-RL, which stands for GRAph neural network-based Scheduler for Production planning problems using Reinforcement Learning. The framework represents job shop scheduling problems (JSSP) as a graph and trains the RL agent using features extracted through a graph neural network (GNN). By leveraging the GNN, the features are extracted in the non-Euclidean space, providing a comprehensive encoding of the current production state in the Euclidean space. The custom message-passing algorithm applied to the GNN plays a crucial role. The node features encoded by the GNN are utilized by the RL agent to make decisions, such as selecting the next job. The scheduling problem is treated as a decentralized optimization problem, where the learning agent is assigned to individual production units, and the agent learns asynchronously from the experience collected on all other production units. GraSP-RL is applied to a complex injection molding production environment with 30 jobs and 4 machines, aiming to minimize the makespan of the production plan. The results show that GraSP-RL outperforms first-in-first-out (FIFO), tabu search (TS), and genetic algorithm (GA) for the task of planning 30 jobs in JSSP. The generalization capability of the trained agent is also tested on two different problem classes: Open shop system (OSS) and Reactive JSSP (RJSSP). In these modified problem classes, GraSP-RL produces results better than FIFO and comparable results to TS and GA without further training, providing schedules instantly.

- Dynamic scheduling for semiconductor manufacturing systems with uncertainties using convolutional neural networks and reinforcement learning [48]

The semiconductor manufacturing systems (SMSs) face increasing complexity and challenges in dynamic scheduling due to internal uncertainties and external demand fluctuations. This paper addresses the integrated release control and production scheduling problems in SMSs with uncertain processing times and urgent orders. The proposed solution is a Convolutional Neural Network and Asynchronous Advanced Actor-Critic-based method (CNN-A3C), which consists of a training phase and a deployment phase.In the training phase, actor–critic networks are trained to predict the evaluation of scheduling decisions and output the optimal scheduling decision. In the deployment phase, the most appropriate release control and scheduling decisions are periodically generated based on the current production status. The authors improve key aspects of the deep reinforcement learning (DRL) algorithm, including the state space, action space, reward function, and network structure. Four mechanisms are designed: a slide-window-based two-dimensional state perception mechanism, an

adaptive reward function considering multiple objectives and adjusting to dynamic events, a continuous action space based on composite dispatching rules (CDR) and release strategies, and actor–critic networks based on convolutional neural networks (CNNs).To validate the proposed dynamic scheduling method, it is implemented on a simplified SMS, and simulation experiments demonstrate its superiority over the unimproved A3C-based method and common dispatching rules, especially in the face of new uncertain scenarios.

- Dynamic production scheduling towards self-organizing mass personalization: A multi-agent dueling deep reinforcement learning approach [69]

This paper introduces a reinforcement learning-based approach for dynamic job shop scheduling problems, employing a static-training-dynamic-execution strategy. The scheduling policies are learned from static scheduling instances using a multiagent dueling deep reinforcement learning approach. The proposed approach includes new representations for observation, action, reward, and cooperation mechanisms between agents. The learned scheduling policies are then applied to a dynamic scheduling system where stochastic processing times and unplanned machine breakdowns can occur randomly. The approach is extensively evaluated through simulation experiments, demonstrating its superiority over heuristic rules (FIFO, SPT, LPT, SNQ, and LNQ) in terms of makespan and handling breakdowns under two dynamic manufacturing settings.

- Design and Implementation of Simulation-Based Scheduling System with Reinforcement Learning for Re-Entrant Production Lines [28]

In this study, a re-entrant production line was simulated as a manufacturing environment, and an adaptive scheduling system was developed to enhance operational performance using deep reinforcement learning (DRL). The study involved creating a software architecture to integrate DRL with the simulation, defining the states, actions, and rewards of the RL agent, and designing a discrete-event simulation control module to collect data and evaluate the trained policy network.Experiments were conducted on a hypothetical re-entrant production line, divided into three cases. Case A compared makespan results with single priority-based dispatching rules, Case B investigated the impact of part sequences, and Case C analyzed flexibility effects. The proposed system showed significant improvements, with the average makespan being 15%, 29%, and 9% smaller than those from FCFS, FOPR, and MOPR rules, respectively.

- Cloud–edge collaboration task scheduling in cloud manufacturing: An attention-based deep reinforcement learning approach [9]

This study addresses cloud–edge collaboration manufacturing task scheduling in Cloud Manufacturing (CMfg) to maximize customer satisfaction and balance production. The proposed approach, named Attention-based Value-function Maximum a posteriori Policy Optimization (AV-MPO), deals with the dynamics and complexity of state information in this context.The Cloud–Edge Collaboration Manufacturing Task Scheduling (CETS) problem is formulated as a partially observable Markov decision process. AV-MPO, employing on-policy maximum a posteriori policy optimization with a gated transformer-XL (GTrXL), is introduced. The algorithm's effectiveness, training stability, generalizability, scalability, and robustness are evaluated. Comparative analysis is conducted against

rule-based algorithms and state-of-the-art DRL algorithms, including proximal policy optimization (PPO), soft actor-critic (SAC), and dueling deep Q network (Dueling DQN).Experimental results demonstrate that AV-MPO effectively addresses the CETS problem, showing maximum improvements of 12.6% for overall scheduling benefit, 13.9% for service rate, and optimal load balance rate in most cases. The algorithm's robustness is validated in unpredictable scenarios, such as device unavailability and service provider outage.

- Application of a Reinforcement Learning-based Automated Order Release in Production [73]

This paper describes the application of a reinforcement learning (RL) algorithm, specifically a Deep Q-Network (DQN), to optimize order release procedures in real-world production scenarios. The focus is on achieving a higher practical orientation, addressing realistic problem sizes, customer orientation, and the development of a control application for performance evaluation.The main objective is to optimize adherence to delivery dates using a DQN algorithm. The study applies this approach to two problem instances: the first with 10 machines and 76 orders, and the second with 10 machines and 259 orders. The results indicate an adherence to delivery dates of 84.21% for the smaller dataset and 91.89% for the larger dataset. Additionally, a larger problem size with 28 machines and 474 orders is explored, highlighting the challenges of direct scaling without adjusting the problem formulation for justifiable training times.

- An improved deep reinforcement learning-based scheduling approach fordynamic task scheduling in cloud manufacturing [83]

"This paper introduces an improved approach for dynamic task scheduling in Cloud Manufacturing (CMfg) using Deep Reinforcement Learning (DRL). The proposed approach addresses issues related to inadequate fine-tuning ability and low training efficiency observed in existing DRL-based scheduling methods. The key contributions include:
1. Identification of causes behind the shortcomings in existing DRL-based scheduling methods.
2. Introduction of a novel approach to address these issues by updating the scheduling policy while considering the distribution distance between the pre-training dataset and the in-training policy.
3. Use of uncertainty weights in the loss function to avoid overestimation of the reward function.
4. Extension of the output mask to the updating procedures.
Numerical experiments on thirty actual scheduling instances demonstrate that the proposed approach outperforms other DRL-based methods (PPO, DRQN, DDQN, A3C, AC, and BC) in terms of solution quality and generalization, with improvements of up to 32.8% and 28.6%, respectively. Additionally, the method effectively fine-tunes a pre-trained scheduling policy, leading to an average reward increase of up to 23.8%.

- A multi-objective reinforcement learning approach for resequencing scheduling problems in automotive manufacturing systems [42]

This study addressed a multi-objective resequencing scheduling problem in automotive manufacturing systems, considering operational requirements in the paint shop and sequential requirements in the assembly shop. Resequencing cars based on color-oriented batches aimed to reduce color change costs and operational costs in paint shops, while assembly shops required timely

completion to ensure high sequence adherence. The study investigated two conflicting objectives - color change costs and sequence tardiness - in a single-machine flowshop scheduling environment. A multi-objective deep Q-network algorithm was developed to determine the Pareto frontier. Reward shaping, 2D-folded-normal distribution for sampling preferences, and other techniques were employed to enhance algorithm performance. Experimental results demonstrated that the proposed approach outperformed meta-heuristic and envelope Q-learning algorithms in terms of solving time, performance, convergence of the neural network, and diversity of the Pareto frontier, making it suitable for improving scheduling efficiency and reducing operational costs in automotive paint shops.

- Distributed Real-Time Scheduling in Cloud Manufacturing by Deep Reinforcement Learning [37]

This study addresses the distributed real-time scheduling (DRTS) of multiple services to meet dynamic and customized orders in cloud manufacturing (CM). The proposed DRTS framework incorporates cloud–edge collaboration, deploying distributed actors in the edge layer and a centralized learner in the cloud layer to enhance performance and responsiveness. The DRTS problem is formulated as a semi-Markov decision process, considering both processing services sequencing and logistics services assignment simultaneously. A distributed dueling deep Q network (D3QN) is developed with cloud–edge collaboration to optimize the weighted tardiness of jobs.Experimental results showcase the effectiveness of the proposed D3QN, demonstrating lower weighted tardiness and shorter flow-time compared to state-of-the-art algorithms. In particular, when compared with three baseline algorithms – GA, HGP, and DQN – the average improvement rates are substantial, with percentages of 35.59%, 28.70%, and 17.33%, respectively, for the former, and 5.29%, 2.66%, and 0.39% for the latter.

- Dynamic job shop scheduling based on deep reinforcement learning for multi-agent manufacturing systems [98]

This paper addresses the challenges posed by personalized orders in the production paradigm by proposing a multiagent manufacturing system based on deep reinforcement learning (DRL). The system integrates self-organization mechanisms and self-learning strategies to enhance dynamic responsiveness and self-adjustment capabilities within the workshop. The manufacturing equipment in the workshop is modeled as equipment agents with support from edge computing nodes. An improved contract network protocol (CNP) guides cooperation and competition among multiple agents, facilitating efficient completion of personalized orders.The decision-making module, named AI scheduler, is established using a multi-layer perceptron within the equipment agent. AI scheduler, informed by perceived workshop state information, generates optimal production strategies for task allocation. Periodic training and updates of AI scheduler occur through the proximal policy optimization (PPO) algorithm, enhancing decision-making performance based on collected sample trajectories of the scheduling process.In the experimental validation within a multiagent manufacturing system testbed, dynamic events such as stochastic job insertions and unpredictable machine failures are considered. Results demonstrate that the proposed method effectively generates scheduling solutions meeting various performance metrics and autonomously handles resource or task disturbances in dynamic events. Comparative analysis against SPT+FIFO, GP-based, and DQN-based methods indicates that the PPO-based method achieves superior solutions in terms

of workload balance, order profit, and evaluation value, particularly under order insertion and machine failure scenarios.

- Solving task scheduling problems in cloud manufacturing via attention mechanism and deep reinforcement learning [86]

This study introduces an end-to-end scheduling algorithm designed to address task scheduling challenges in computer-integrated manufacturing (CMfg). The proposed algorithm utilizes the multi-head attention mechanism to capture intercorrelations within the enterprise–enterprise and enterprise–task relationships and is trained using Deep Reinforcement Learning (DRL). Notably, the proposed algorithm demonstrates remarkably low response times compared to heuristic algorithms, providing scheduling solutions within seconds.Unlike other DRL algorithms, the proposed approach exhibits improved scheduling performance and adopts a more accessible modeling method. It achieves stability in training without the necessity for a step-based reward function, relying solely on the objective function. The incorporation of multi-head attention and DRL into scheduling problems represents an exploratory effort, yielding positive outcomes.Experimental results, conducted on a case involving the processing of automobile structure parts in CMfg, indicate that the proposed algorithm exhibits competitive scheduling performance and runtime compared to eight DRL algorithms, two heuristic algorithms, and two priority dispatching rules. Furthermore, the proposal demonstrates superior generalizability and scalability when compared to the other eight DRL algorithms, specifically SAC, PPO, DDQN, DQN, DQN with fixed Q-targets, Dueling DDQN, A3C, and A2C.

- Dynamic scheduling of tasks in cloud manufacturing with multi-agent reinforcement learning [85]

This paper introduces a novel scheduling algorithm, MAGCIS (Multi-Agent Graph Convolution Integrated Scheduler), designed to address dynamic scheduling challenges in the group service cloud manufacturing (GSCMfg) environment. The algorithm is formulated and trained using multiagent reinforcement learning.MAGCIS incorporates graph convolution to encode the graph-structure features of tasks, and a recurrent neural network is employed to record the processing trajectories of each task. The algorithm is trained with a mixing network under a centralized training decentralized execution architecture. The action space and reward function are independently designed.In a case study focused on aircraft structural part processing, MAGCIS demonstrated superior performance and generalizability compared to six other multi-agent reinforcement learning algorithms (QMIX, VDN, QTRAN_alt, QTRAN_base, REINFORCE, and Central_Vin). The paper suggests that MAGCIS has the potential to be applied to scheduling environments similar to GSCMfg, providing detailed insights into its training and execution processes.

- Using real-time manufacturing data to schedule a smart factory via reinforcement learning [18]

Utilizing real-time manufacturing data, this paper aims to design a dynamic scheduling method for the efficient production of a smart factory. The proposed approach integrates a Multi-Agent System (MAS)-based dynamic scheduling mechanism with a double Q-Learning algorithm. The dynamic

scheduling mechanism begins with the design of the problem formulation module and scheduling point module. Subsequently, a genetic programming (GP) method is employed to generate sixteen high-quality rules, forming the scheduling rule library. Additionally, a state clustering module is introduced, utilizing autoencoder, self-organizing mapping neural network, and k-means clustering algorithm to efficiently cluster production attribute vectors. In the decision-making process, an improved Q-learning algorithm is applied to train the GP rule selector. This empowers the decision-making agent to select the appropriate GP rule based on the production state at each scheduling point. Experimental results demonstrate the feasibility and superiority of the proposed method in real-time scheduling. The approach exhibits effectiveness in handling disturbance events within the manufacturing process, showcasing its potential for optimizing smart factory production objectives.

- Multi-Agent Reinforcement Learning for Real-Time Dynamic Production Scheduling in a Robot Assembly Cell [12]

A Multi-Agent Reinforcement Learning (MARL) system is introduced for the dynamic scheduling of assembly jobs in a robot assembly cell. The approach employs a Double Deep Q-Network (DQN) algorithm and introduces a generalized observation, action, and reward design tailored for the dynamic flexible job shop scheduling (FJSP) context.During a centralized training phase, each agent (robot) within the assembly cell makes decentralized scheduling decisions based on local observations. The proposed solution consistently achieves shorter makespans, enhancing the overall efficiency of the robot assembly cell. The algorithm's validation is demonstrated through a conveyor case study, with the MARL system design being broadly applicable to commonly studied FJSP scenarios. The study also explores the impact of varying observation sizes for each agent on optimization performance.

- Deep reinforcement learning based scheduling within production plan in semiconductor fabrication [40]

This study employs deep reinforcement learning (RL) to address scheduling processes within a production plan. The Deep Q-network (DQN) algorithm is utilized, and novel state, action, and reward definitions are introduced to optimize the scheduling policy. The performance of the proposed deep RL method is compared with other dispatching rules, demonstrating its superiority across diverse cases. Particularly, the study focuses on a semiconductor fabrication model, where the DQN algorithm is compared to Setup-based, PBB, and Plan-based rules. The proposed method achieves an approximately 19% to 21% improvement in average throughput compared to the dispatching rules. Furthermore, the average lead-time of the proposed method decreases by approximately 39% to 63% in comparison to the dispatching rules.

- Reinforcement learning for online optimization of job-shop scheduling in a smart manufacturing factory [100]

This paper introduces a smart scheduler designed to manage real-time jobs and unexpected events within smart manufacturing factories. The smart scheduler employs composite reward functions to enhance decision-making capabilities and learning efficiency. Utilizing deep reinforcement learning (RL), the scheduler autonomously learns to schedule manufacturing resources in real-time,

dynamically improving its decision-making abilities. The proposed scheduling model is evaluated through experiments on a smart factory testbed. Results demonstrate that the smart scheduler, optimized with composite reward functions, achieves efficient learning and scheduling performances, effectively handling unexpected events such as urgent or simultaneous orders and machine failures. In comparison to common online or offline scheduling methods, RL-based scheduling with composite rewards (RL-C) outperforms Genetic Algorithm (GA), Shortest Processing Time First (SPTF), and First Come First Serve (FCFS) methods. FCFS, relying on specific rules, lacks the ability to enhance decision-making during scheduling processes. RL-based methods, specifically RL-B and RL-C, initially focus on minimizing order waiting times in real time. RL-C converges faster than RL-B, requiring 37.0% fewer training episodes. Thus, the proposed RL-C scheduling method demonstrates superior learning performances compared to traditional rule-based and basic RL scheduling methods.

- Reinforcement learning approach to scheduling of precast concrete production [33]

This study introduces a precast concrete (PC) production scheduling model based on a reinforcement learning approach, offering the flexibility to address various problem conditions with rapid computation and real-time efficacy. Experimental results reveal that the proposed model consistently outperforms other methods, demonstrating a 4–12% improvement in total tardiness with an average winning rate of 77.0%. The model holds the potential to enhance the success of off-site construction projects by ensuring stable progress in PC construction. Q-learning, along with dispatching rules like EDD, CR, SPT, and FIFO, was applied in the same case to highlight the effectiveness of the proposed model. The results indicate that the proposed model achieves the best total tardiness value of 20.5 by selecting dispatching rules based on shop conditions. EDD follows as the second-best method (31.0), with Q-learning ranking third (37.6). CR and SPT exhibit lower performance, while FIFO performs the least effectively in the practical case.

- Task Allocation in Human–Machine Manufacturing Systems Using Deep Reinforcement Learning [30]

Despite the increasing prevalence of automation in manufacturing systems, human operators remain essential for various activities. This work introduces a framework for task allocation in human–machine manufacturing systems, employing a reinforcement learning (RL)-based method. The agent is trained iteratively in an RL framework using task allocation data, and recurrent layers are used to assess and enhance unobservable states of human operators. The agent allocates tasks based on the expected cumulative discounted reward in each episode, considering factors such as fatigue accumulation and task competence level of human operators. The proposed approach, utilizing deep learning (DL) as a framework and RL for performance updates, outperforms classical dispatching rules in terms of mean flowtime. In comparison to shortest processing time (SPT) and first-in-first-out (FIFO) rules, the proposed method yields a shorter mean flowtime with less variation. Specifically, for 1500 episodes, the proposed method demonstrates a mean and standard deviation of mean flowtime ($m = 25.97$, $s = 16.44$), outperforming the SPT rule ($m = 39.98$, $s = 29.19$) and FIFO rule ($m = 39.84$, $s = 24.47$). This outcome suggests improved performance in terms of flowtime, attributed to workload balance among operators and task assignment based on appropriate competence levels.

- A Dynamic Chemical Production Scheduling Method based on Reinforcement Learning [95]

This paper addresses the challenge of dynamic chemical production scheduling, where processing strategies need to adapt to changing order demands. Traditional methods struggle with the uncertainty of task objectives in this context. The paper introduces proximal policy optimization algorithms, a type of reinforcement learning method, to tackle this issue. An improved state function, considering the difference between short-term and long-term orders, is proposed. This enhancement effectively resolves the dynamic chemical production scheduling problem with uncertainty. Experimental results on the dynamic chemical production scheduling model, compared with the policy gradient algorithm, demonstrate that the proposed method achieves higher rewards in scheduling, faster convergence, and less performance fluctuation. Specifically, the mean profit and mean variance in the Policy Gradient case are 181.3481 and 686.4350, respectively, while in the PPO case, they are 203.5996 and 45.0526, respectively. These results indicate that the methods presented in this paper can consistently handle the complexity and uncertainty of chemical production scheduling.

- Reinforcement learning and digital twin-based real-time scheduling method in intelligent manufacturing systems [96]

This work introduced a novel approach to real-time scheduling, named "twins learning," which integrates reinforcement learning and digital twin technology. This innovative method aims to address diverse objectives concurrently. Initially, a virtual twin is created to simulate the interaction among multiple resources, encompassing physical aspects, behaviors, and rules essential for decision-making. Subsequently, the real-time scheduling challenges are formulated as a Markov Decision Process, and dedicated reinforcement learning algorithms are crafted to acquire improved scheduling

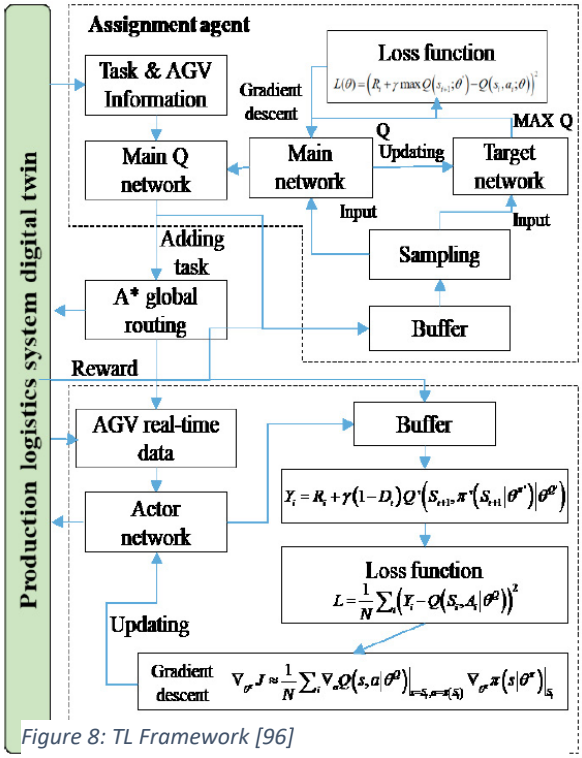

Figure 8: TL Framework [96]

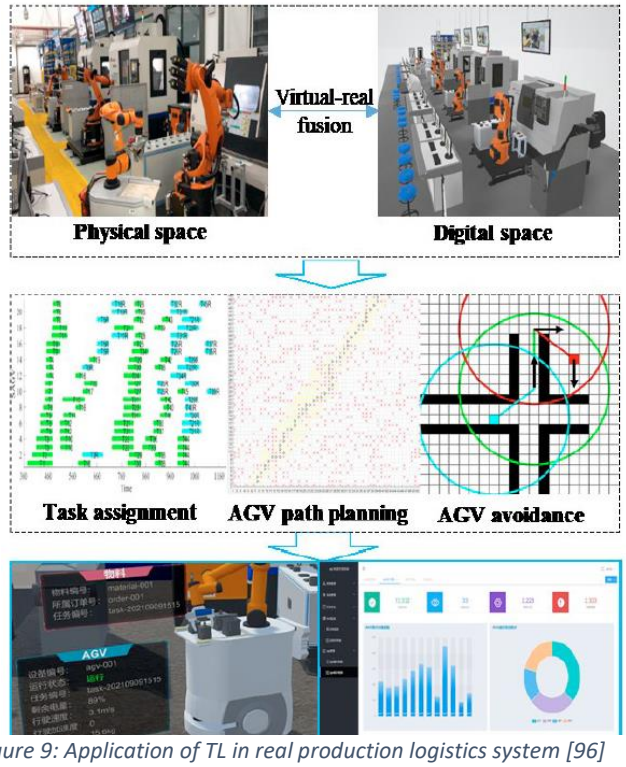

Figure 9: Application of TL in real production logistics system [96]

policies. The outcomes of a case study demonstrate the exceptional adaptability and learning capabilities of the proposed method in the realm of intelligent manufacturing. Leveraging the Markov Decision Process (MDP) framework for both the task assignment problem and the AGV (Automated Guided Vehicle) avoidance problem, they have developed specialized agents for task assignment and avoidance. These agents are designed to achieve optimal task assignment and conflict-free routing. The assignment agent is dedicated to optimizing the allocation of tasks, while the avoidance agent focuses on ensuring AGV paths are free from conflicts. This integrated approach harnesses the power of MDP to enhance efficiency and streamline operations in the context of task assignment and AGV routing.

- Discovery of customized dispatching rule for single-machine production scheduling using deep reinforcement learning [10]

"In this research, they applied deep reinforcement learning to the optimization of dispatching rules tailored to specific production states in the context of a single-machine production scheduling problem. To formulate these customized dispatching rules, the authors leveraged parameters inherent to the production system (Processing time pk, Batch size of the machine b. Due date Dk and Quantity qk). Each rule was crafted by multiplying individual parameters with random weights, incorporating arithmetic operators, and determining whether the largest or smallest numerical value linked to the production order should be chosen. Recognizing the impracticality of storing Q values using a traditional Q-table due to complexity, a deep Q network was employed to calculate Q-values based on four key attributes representing the production state. These attributes include the number of queuing production orders, mean and standard deviation of slack time for all queuing production orders, and the batch size of the machine.
The primary production objective, measured in terms of total tardiness, guided the evaluation. Preliminary results indicated the effectiveness of the customized dispatching rules selected through deep reinforcement learning across a majority of the test cases. Moreover, the approach outperformed traditional scheduling algorithms such as FIFO, EDD, and SPT, showcasing its potential for enhanced production scheduling outcomes. Three widely recognized dispatching rules - FIFO, SPT, and EDD - were also employed in the production process. Notably, there were instances where SPT exhibited superior performance in minimizing total tardiness (three instances). The current study, conducted over 500 training episodes, anticipates that augmenting the number of training episodes could further enhance production performance, specifically in terms of minimizing total tardiness, by dynamically assigning customized dispatching rules. Upon comparing the results of these established dispatching rules with those of the dynamically assigned customized dispatching rules, preliminary findings indicate that in 46 out of 50 instances (92% of all tested cases), the dynamic assignment of customized dispatching rules led to better performance in total tardiness minimization. This underscores the potential effectiveness and adaptability of dynamically assigning dispatching rules based on the evolving production state.

- Deep Reinforcement Learning-Based Job Shop Scheduling of Smart Manufacturing [101]

To address the evolving challenges of scheduling operations on machines, this paper introduces a novel approach, employing Deep Reinforcement Learning with an Actor-Critic algorithm (DRLAC). The Job-Shop Scheduling Problem (JSSP) is formulated as a Markov Decision Process (MDP). The state of a

JSSP is represented using Graph Isomorphism Networks (GIN) to extract node features during scheduling. The optimal scheduling policy is derived by guiding these included node features to the best next action in the schedule. The training algorithm of the Actor-Critic (AC) network, based on reinforcement learning, is adopted to achieve the optimal scheduling policy. To demonstrate the effectiveness of the proposed model, a case study is presented illustrating a conflict between two job schedules. Furthermore, the proposed model is applied to a benchmark dataset, and the results are compared with traditional scheduling methods and trending approaches. Numerical results indicate that the proposed model exhibits adaptability in real-time production scheduling. The average percentage deviation (APD) of our model achieves values between 0.009 and 0.21 when compared with heuristic methods, and values between 0.014 and 0.18 when compared with other trending approaches (GA DRL MARL DDPG DRLAC APD-DRLAC). These findings suggest the efficacy of the proposed model in optimizing scheduling operations.

- An Adaptive Reinforcement Learning-Based Scheduling Approach with Combination Rules for Mixed-Line Job Shop Production [102]

This paper introduces an adaptive real-time scheduling method tailored for the mixed-line job shop scheduling problem, incorporating combined processing constraints. The innovation lies in the introduction of a virtual operation, effectively simplifying and transforming the challenge posed by combined processing constraints into a classical flexible job shop scheduling problem. To address the dynamic coexistence of trial-production and batch production scenarios, a disturbance processing mechanism is established. Recognizing the significant impact of emergency trial-production orders on batch production plans, the k-nearest neighbor method is employed to identify historical operations most similar to trial-production components. To overcome the limitations of traditional single dispatching rule strategies, the scheduling decision-making process is divided into the machine selection stage and buffer job sequencing stage. A scheduling decision model is established based on contextual bands (CBs) within reinforcement learning, employing rough continuous trial and error learning. This enables each scheduler to dynamically select optimal machine selection rules and buffer job sequencing rules based on the real-time state of the scheduling environment, significantly enhancing adaptability and performance. The proposed methodology is evaluated and validated through experiments conducted in a smart manufacturing setting, the mixed-line job shop of missile structural parts in Shanghai. Results demonstrate that the proposed algorithm achieves a 5% and 2% performance improvement in completion time compared with epsilon greedy and Q-learning algorithms, respectively. Furthermore, the proposed method exhibits more than a 10% improvement even compared to the best rule (SPT + FIFO) among the nine single dispatching rules in this simulation experiment (SQ+FIFO, SQ+SJF, SQ+LIFO, LQE+FIFO, LQE+SJF, LQE+LIFO, SPT+FIFO, SPT+SJF, SPT+LIFO). When compared with epsilon greedy and Q-learning algorithms, the proposed algorithm demonstrates a 7% and 5% performance improvement in terms of completion time.

- A fuzzy hierarchical reinforcement learning based scheduling method for semiconductor wafer manufacturing systems [82]

This paper introduces a Fuzzy Hierarchical Reinforcement Learning (FHRL) approach for scheduling in a Semiconductor Wafer Fabrication System (SWFS). The primary objective is to control the cycle time (CT) of each wafer lot, enhancing on-time delivery by adjusting the priority of each lot. The proposed

FHRL approach addresses challenges arising from layer correlation and wafer correlation of CT due to re-entrant process constraints. The hierarchical model incorporates a recurrent reinforcement learning (RL) unit in each layer to control the corresponding sub-CT of each integrated circuit layer. Within each RL unit, a fuzzy reward calculator is designed to mitigate the impact of uncertainty in expected finishing times resulting from the rematching of a lot to a delivery batch. The results demonstrate the efficacy of the FHRL approach, with the mean deviation (MD) between actual and expected completion times of wafer lots under its scheduling being only about 30% of the compared methods across the entire SWFS(Ada_Rule, GEP,FHRL, FIFO, SPT, and EDD).This suggests the superiority of the proposed FHRL approach in achieving more accurate and reliable scheduling outcomes in semiconductor wafer fabrication.

- Dynamic matching with deep reinforcement learning for a two-sided Manufacturing-as-a-Service (MaaS) marketplace [61]

This paper addresses the challenge of near-real-time decision-making for suppliers within a manufacturing-as-a-service (MaaS) marketplace. Traditional myopic decision-making methods, like a first-come, first-serve approach, may result in suboptimal revenue generation in this dynamic and stochastic environment. The problem is formulated as a Markov Decision Process (MDP) and tackled using deep reinforcement learning (DRL). The empirical simulations conducted in the study compare the performance of DRL with four baselines (TQ, Random, GH, and RHA). In the simulated environment, orders arrive sequentially on the platform, and in each period, the platform takes actions to accept orders until it selects an invalid order or the action "wait." The simulator updates the environment based on the accepted orders and moves to the next period. Waiting orders remain in the queue until a period before their due date. Results indicate that DRL, specifically the Deep Q-Network (DQN), outperforms the baselines considerably. DQN demonstrates a higher order acceptance rate, attributed to its ability to learn, and maximize revenue over time. It efficiently manages capacity, choosing smaller orders with higher revenue per hour and reserving capacity for potentially better orders in the future. The baseline TQ performs slightly better than RHA but significantly worse than DQN, primarily due to the limited environmental information captured in its state definition. In conclusion, this early work showcases a learning approach for near-real-time decision-making in a MaaS marketplace, highlighting the effectiveness of DRL, particularly the DQN model, in optimizing order acceptance and revenue generation for suppliers.

- Reconfigurable manufacturing system scheduling: a deep reinforcement learning approach [78]

This paper addresses the challenges and opportunities posed by Reconfigurable Manufacturing Systems (RMS) in efficiently scheduling multiple products. A novel approach is presented, utilizing a dynamic control policy established by a group of deep reinforcement learning agents. These collaborative agents, equipped with a shared value decomposition network, work towards minimizing the make-span by guiding automated guided vehicles to transport modules of machines, raw materials, and finished products within the RMS.To evaluate the effectiveness of this framework, two numerical case studies are implemented. Both case studies involve a fully automated RMS tasked with producing three different products. The results from these case studies indicate that the proposed training framework is applicable and efficient, especially for medium and small-scale RMS scheduling

problems, as long as a substantial number of training iterations are conducted before the algorithm starts to perform effectively.

- Designing an adaptive production control system using reinforcement [35]

In the pursuit of operational excellence within the competitive landscape of manufacturing companies, traditional production control methods are proving insufficient. To address this, the paper introduces a methodology for the design of an adaptive production control system utilizing reinforcement learning. The application of reinforcement learning in production control offers an alternative approach, especially with the increasing digitization of manufacturing processes. The methodology addresses key challenges in the application of reinforcement learning methods: Designing State Information - the information provided to the RL agent as the state greatly influences its learned performance and Modeling the Reward Signal - the formulation of the reward signal is crucial as it represents the optimization objective. The paper thoroughly explores and applies these principles to two real-world production scenarios from the semiconductor industry. The analysis reveals the adaptability of RL agents to different objectives and application scenarios. Even "simple" RL agents with a constant reward function outperform random heuristics. Specific RL agents can surpass existing rule-based benchmark heuristics (RANDOM, FIFO, and VALID). Additionally, an enhanced state representation improves performance when related to the objectives. The design of the reward signal allows for easier optimization of multiple objectives. Lastly, specific configurations of RL agents exhibit high performance across different production scenarios. In conclusion, the study demonstrates that reinforcement learning can be successfully applied to achieve adaptive control strategies in manufacturing, showcasing its flexibility and superior performance compared to traditional methods in certain scenarios.

- Control of Shared Production Buffers: A Reinforcement Learning Approach [56]

This study addresses a buffer control problem inherent in stochastic flow lines featuring shared production buffers. Buffer control involves the implementation of decision rules governing the movement of items between buffers and machines, particularly during the release or completion times of parts across different production stages. The authors present a conceptual model for this problem, focusing on a fundamental scenario with a central buffer. They elucidate how this model can be extended to encompass various system configurations, ultimately addressing a tactical buffer allocation problem. If the flow line exhibits characteristics of a Markovian production system, the authors formulate the problem as a continuous-time Markov decision problem, seeking an optimal stationary policy. To facilitate the solution, they employ a uniformization approach from the literature, enabling the discretization of the Markov decision problem in time and making it amenable to standard algorithms. The paper proposes a straightforward implementation of Q-learning, a reinforcement learning technique, which converges to an optimal stationary policy. The effectiveness of this approach is validated through a numerical experiment involving a small-scale toy problem.

- A reinforcement learning model for material handling task assignment and route planning in dynamic production logistics environment [29]

This study aimed to assess the application of Reinforcement Learning (RL) in material handling tasks within a Shared Production Line (SPL) context in manufacturing companies. Specifically, the focus was on the routing of Automated Guided Vehicles (AGVs) for material handling, taking dynamic aspects into account. The study introduced an architecture that integrates RL into SPL, defining key elements of RL such as environment, value, state, reward, and policy. The managerial implications of the findings suggest a departure from traditional fixed-route policies for material handling. Instead, applying RL in SPL can lead to dynamic and real-time assignment, sensing, and response to individual needs in material handling, potentially reducing makes pan, distance, and energy consumption while enhancing the overall responsiveness of SPL, thereby contributing to increased manufacturing competitiveness. The model presented in this study is based on the Q-learning algorithm. The main goal of Q-learning is to formulate and establish a policy (denoted as "p") for guiding the Automated Guided Vehicle (AGV) effectively along the optimal path. Through extensive exploration of numerous possibilities for material transfer, the AGV, functioning as an agent, is expected to learn and adopt a well-defined deterministic policy that outlines the appropriate course of action in any given situation. Initially, the approach involved the implementation of a completely random policy. However, as the model underwent learning from thousands of potential navigation scenarios, it gradually refined and improved its performance.

- Integrated Planning and Scheduling for Customized Production using Digital Twins and Reinforcement Learning [55]

To tackle the challenge of customized production, the paper introduces a self-learning process planning approach based on a digital twin, utilizing Deep-Q-Network. This method is designed to identify optimized process plans and workflows for the concurrent production of personalized products. The authors conducted an evaluation of the approach using a virtual aluminum cold milling factory from the SMS Group within the BaSys 4 project context. The objective of the evaluation was to demonstrate that the proposed approach is effective in managing a large problem space. The approach to integrated planning and scheduling introduces a broader problem space, necessitating the utilization of advanced reinforcement learning techniques, such as the Deep Q-Network (DQN), to discover nearly optimal solutions within a feasible timeframe. The methodology is crafted for seamless integration into a production system based on a Service-Oriented Architecture (SOA). Within this framework, Production Decision Tables (PDTs) and Resource Decision Tables (RDTs) serve as representations of assets during the production phase, actively collecting authentic production data. In the event of unforeseen disruptions requiring rescheduling, PDTs and RDTs transform into virtual products and resources, actively engaging in the integrated planning and scheduling processes. This dynamic involvement allows the approach to access real-time data from the operational factory, facilitating decision-making grounded in authentic production conditions.

- Modeling Production Scheduling Problems as Reinforcement Learning Environments based on Discrete-Event Simulation and OpenAI Gym [39]

In this study, a methodology was introduced to guide the modeling of production scheduling problems as Reinforcement Learning (RL) environments. The approach involved the integration of Discrete Event Simulation (DES) and the OpenAI Gym interface. DES was employed for modeling the underlying processes and dynamics inherent in any scheduling problem, while the OpenAI Gym

interface ensured a standardized development process, facilitating the deployment of pre-implemented RL algorithms. This method enabled the deployment of algorithms and agents for two distinct scheduling problems: the allocation of jobs to resources and the sequencing of jobs for a resource. Additionally, the proposed step method provided flexibility, allowing for the training of a single agent for one of the two problems or multiple agents simultaneously for both problems.

- Simultaneous Production and AGV Scheduling using Multi-Agent Deep Reinforcement Learning [68]

The growing utilization of automated guided vehicles (AGVs) introduces additional flexibility and complexity to overall production systems, leading to the emergence of the Flexible Job Shop Scheduling Problem (FJSSP). This problem, involving the coordination of AGVs, is NP-hard and challenging to optimize. To address this, a Reinforcement Learning Multi-Agent (MARL) system is proposed, where job scheduling, and vehicle planning are collaboratively managed. The concept is outlined and implemented in a prototype. Experiments were carried out in a Unity-based simulation environment with 4 production cells and 2 AGVs, solving randomly generated production planning problems comprising 13 jobs of types 1-3. The theoretical minimum production time, including final transport to the outgoing warehouse, is 554 time units. As a reference, Shortest-Job-Next (SJN) and Earliest Due Date (EDD) heuristics are employed for the machines in the multi-agent system. The RL-based agent system outperforms the heuristics, achieving up to a 100-time unit improvement. While the results surpass the minimum of 554 time units for all processes, constraints such as transport capacity limits and physical location pose challenges, highlighting the importance of effective coordination between machines and logistics.

- A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence [90]

In this study utilizes digital twin simulation and communication technologies to construct virtual counterparts of robot manufacturing systems. These virtual environments serve as the foundation for training an intelligent scheduler based on Deep Reinforcement Learning (DRL) in a safe and controlled manner. Unlike previous attempts at integrating Reinforcement Learning into work cell scheduling, the proposed system-level digital twinning extends to complex manufacturing systems, leveraging deep neural networks to address challenges related to large state and action spaces.The implementation involves the creation of high-fidelity Virtual Commissioning platforms using Siemens Tecnomatix Process Simulate. These platforms simulate and synchronize system components with live signals, employing advanced tools for event-based simulation, collision detection, robot reachability testing, and robot configurations through reverse kinematics. The offline programming process enables the direct transfer of generated robot programs to physical robot systems without intermediate translations. After constructing the virtual environment, system communications are implemented on both virtual and physical pathways. "Software-in-the-loop" and "Hardware-in-the-loop" testing methods are discussed, serving as the baseline for virtual commissioning control loops. The intelligent scheduler's communication pathway with the virtual cell is established, facilitating repetitive offline training cycles and enabling Industrial Internet of Things (IIoT) for remote human intervention through customized OPC clients. The Manufacturing Intelligence algorithm, framed by Deep Reinforcement Learning, is trained on the constructed digital twin. Both natural Deep Q-

Learning and its enhanced version, incorporating Prioritized Experience Replay and Double Q Network, are implemented to improve data efficiency. The outcome is a robust dynamic scheduler, trained as Deep Reinforcement Networks, capable of being fed by live signals from either the physical cell or its digital twin. These networks are designed to be reusable and transferable for other specific learning tasks. The integration of the proposed Digital Engine that supports scheduling in an industrial virtual commissioning platform significantly enhances the capabilities of data analytics by interfacing with industrial simulation and automation software. This data-driven manufacturing intelligence is poised for deployment in specific industrial applications, exemplifying a use case for Smart Manufacturing implementation.

- A Deep Reinforcement Learning Based Scheduling Policy for Reconfigurable Manufacturing Systems [77]

"To minimize reconfiguration actions in a generic Reconfigurable Manufacturing System (RMS), this paper employs a deep reinforcement learning agent in conjunction with a built-in discrete event simulation model. The agent, focused on completing assigned order lists while minimizing reconfiguration actions, exhibits superior performance compared to the conventional first-in-first-out dispatching rule after self-learning. The use of Deep Q-Network (DQN) scheduling agents demonstrates a significant potential advantage in this context. The study reveals that DQN agents outperform the traditional dispatch rule (first-in-first-out) even with a limited number of simulations turns. Training results indicate that various DQN agents can quickly converge to an above-average level. Dueling Double DQN (DDDQN) agents, known for their advanced stability, exhibit improved stability as suggested by their inventor. However, it is noted that, despite the enhanced converging capability and stability of DDDQN compared to basic DQN, there are instances where an agent may deviate significantly from optimal performance, emphasizing the need for ongoing optimization of agent stability using advanced techniques like priority replay memory and bagging strategy.
The observation that all agents converge in several episodes suggests that the oversimplified nature of the RMS in the simulation may not fully unveil the potential of the agents. The model's simplicity, with only the initial configuration of every Reconfigurable Machine Tool (RMT) introducing randomness, indicates that future models should consider more realistic factors. These factors may include fluctuated delivery times, random breakdowns of RMTs, a limited number of modules, material resources, products with multiple manufacturing processes, and dynamic order lists with task insertions. Addressing these complexities will contribute to a more comprehensive and representative simulation model.

- Reinforcement Learning With Composite Rewards for Production Scheduling in a Smart Factory [76]

This paper introduces an AI scheduler designed for online and dynamic scheduling of manufacturing jobs within a smart factory. The reinforcement learning (RL) method incorporated into the system imparts self-organizing and self-learning capabilities, particularly under uncertain conditions. A novel composite reward function is formulated to enable the AI scheduler for multi-objective learning and optimization of production schedules, considering rewards for time savings (RD), energy profits (RP), machine utilization (RU), and workload distribution (RV). The proposed AI scheduler relies on a manufacturing value network to estimate state-action values using high-dimensional sensor data from

manufacturing components. It then learns real-time policies based on the states of available machines and pending jobs. The intelligence of AI schedulers is enhanced through streamed data feeds, training experiences, and online learning, demonstrating potential for generalizing to new work orders. The method is capable of handling simultaneously created work orders and uncertainties, such as machine failures. The composite reward function effectively addresses urgent work orders while maintaining a balance between efficiency and profits. The methodology is evaluated in a smart manufacturing setting, demonstrating improved multi-objective performance metrics and effective coping with unexpected events like urgent work orders and machine failures. Simulation results comparing make spans for different scheduling methods (i.e., optimum, AI, RL, and CNP) under normal conditions and machine failures show the AI scheduler's comparable performance with optimal solutions, with only a minor time delay in case of machine failures.

- Two-time scale reinforcement learning andapplications to production planning [97]

This paper centers around the application of a two-time-scale reinforcement learning (RL) approach, utilizing a production planning system as an illustrative example to showcase its concepts and initial outcomes. Monte Carlo simulations serve as the data source for training and validation purposes. The primary objective is to highlight the effectiveness of the two-time-scale method in reducing dimensionality, thereby addressing system complexity. The results presented in this paper demonstrate favorable approximations, particularly when the underlying process exhibits both weak and strong interactions. A production planning problem with failure-prone machines is used throughout this study to illustrate the main ideas, key steps and results.

- Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system [34]

"In this study, a smart manufacturing system was introduced, featuring machines equipped with intelligent agents possessing autonomy in decision-making, sociability for interaction, and the ability to dynamically learn from changing environments using Multi-Agent Systems (MAS) and Reinforcement Learning (RL). The unique aspects of this study compared to existing ones in the field include:
i) Application of autonomous distributed decision-making of machines to a scheduling problem.
ii) Implementation of a method to assess the importance of jobs by dynamically learning in changing environments.
iii) Inclusion of functions enabling machines to voluntarily drop themselves in job negotiations for the overall system's benefit.
iv) Implementation of a mechanism where a specific machine requests other machines to drop out during job negotiations.
v) Integration of learning, adaptation, and decision-making in response to dynamically changing environments by applying RL to the functions mentioned in iii) and iv).
The significance of the proposed system, as indicated by experimental results, includes the observed effectiveness of evaluating job importance, especially in sequence-dependent setup time environments. Additionally, performance comparisons with dispatching rules (SPT, EDD, LPT, and LIS) and a benchmarking distributed MAS algorithm show that the proposed Smart Manufacturing System (SMS) is effective in scheduling for a personalized production system.

- A Reinforcement Learning Approach to Robust Scheduling of Semiconductor Manufacturing Facilities [26]

This article addresses a scheduling problem in the die attach and wire bonding stages of a semiconductor packaging line, considering variabilities in production requirements, available machines, and initial setup status. The study proposes a novel scheduling method using reinforcement learning to enhance robustness and achieve performance improvements. To test the robustness, neural networks trained on small-scale problems are applied to solve large-scale scheduling problems. Results demonstrate that the proposed method outperforms existing approaches (GA, SSU, SPTSSU, MOR, MWR, and SPT) with a short computation time. Moreover, the trained neural network performs well on unseen real-world scale problems, suggesting the method's viability for actual semiconductor packaging lines.

- Intelligent scheduling of discrete automated production line via deep reinforcement learning [74]

This paper introduces a deep reinforcement learning (RL)-based online scheduling method for discrete automated production lines. The paper establishes a discrete event simulation (DES) environment to provide an intelligent and efficient setting for the RL model, achieving competitive performance in online intelligent scheduling policies. The proposed method includes a state modeling approach for discrete automated production line processing, simplifying state complexity and enhancing the precision of the simulation environment. The intelligent scheduling based on deep RL combines DES and iterative learning of RL. The agent is provided with ample opportunities to continually choose transferring actions by introducing null events after transferring. The algorithm effectively handles conflicts between driving simulation time and the occurrence of the next event, considering the real time consumed by transferring. The intelligent scheduling based on deep RL demonstrates the ability to learn efficient policies in various production lines adaptively and exhibits robustness to processing time randomness. In scenarios where processing time follows a log-normal distribution, the agent learns to schedule effectively in different randomness levels by adjusting variances. Experiments conducted on linear, parallel, and re-entrant discrete automated production lines validate that the scheduling policies have comparable performance to heuristic scheduling. Deep RL strikes a balance between efficiency and stability in linear and re-entrant scenarios, although it may perform less optimally than heuristics in parallel scenarios. Additionally, a comparison between deep RL and FIFO in stochastic scenarios indicates that deep RL exhibits sufficient robustness to random processing time.

- Reinforcement learning for an intelligent and autonomous production control of complex job-shops under time constraints [5]

Reinforcement learning (RL) presents promising opportunities for addressing the increasing complexity in managing modern production systems. The authors of this study employ a Q-learning algorithm in conjunction with a process-based discrete-event simulation to train a self-learning, intelligent, and autonomous agent for the order dispatching decision problem in a complex job shop environment with strict time constraints. This work is the first to combine RL in production control

with strict time constraints, and the simulation accurately represents the characteristics of complex job shops commonly found in semiconductor manufacturing. The study addresses a real-world use case from a wafer fab, and the developed and implemented framework is evaluated against benchmark heuristics. The results indicate that RL can be successfully applied to manage order dispatching in a complex environment with time constraints. The RL agent, equipped with a gain function that rewards the selection of the least critical order concerning time constraints, outperforms heuristic rules that strictly follow the selection of the most critical lot first. Consequently, this research demonstrates that a self-learning agent can effectively manage time constraints, with the RL agent performing better than the traditional benchmark, a time-constraint heuristic that combines due date deviations and a classical first-in-first-out approach.

- Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network [23]

To address the dynamic scheduling challenges in Flexible Manufacturing Systems (FMSs), which involve shared resources, route flexibility, and stochastic arrivals of raw products, this paper presents a novel Petri-net-based dynamic scheduling approach utilizing Deep Q-Networks (DQN) with Graph Convolutional Networks (GCN).The timed Stochastic Sequential Process Resource (S3PR) is employed to model an FMS, capturing operation sequential order, resource utilization constraints, and processing time. Subsequently, a Petri-Net-based Convolutional (PNC) layer is designed, incorporating two graph convolution sub-layers. These sub-layers facilitate feature propagation from places to transitions and transitions to places, respectively. The PNC layer offers the advantage of having trainable parameters related solely to the number of filter channels, overcoming the parameter explosion problem associated with building deep neural networks. Finally, a masked DQN, integrated with the PNC network, is utilized to address the timed S3PR dynamic scheduling problem defined by the Markov Decision Process (MDP). Three experiments were conducted to validate the efficacy of the proposed method. The first experiment demonstrated that the simple PNC network effectively handles timed S3PR states with improved stability compared to a Multi-Layer Perceptron (MLP). The second experiment illustrated that the proposed masked DQN with the PNC network achieves comparable dynamic scheduling performance with significantly faster online computation compared to heuristic search methods (D2WS and FCFS+). Moreover, the scheduling performance, learning convergence, and robustness of the proposed method were found to be superior to MLP-based methods. In the third experiment, the adaptability of the proposed method to environmental changes surpassed that of heuristic methods.

- Logistics-involved QoS-aware service composition in cloud manufacturing with deep reinforcement learning [46]

In this work, they introduce a novel Quality of Service (QoS)-aware service composition model for cloud manufacturing, leveraging Deep Reinforcement Learning (DRL) with a specific focus on integrating logistical considerations. The approach involves the development of a DRL-based service composition model tailored for cloud manufacturing, explicitly addressing logistical challenges. The core of the model involves the formulation of the service composition process as a Markov Decision Process, a mathematical framework suitable for decision-making scenarios. To enhance the model's ability to navigate logistical complexities, they crafted a reward function that incorporates logistical

considerations. The DRL algorithm employed in the study, denoted as PD-DQN, amalgamates key components such as the foundational DQN algorithm, the dueling architecture, and the prioritized replay mechanism. This algorithm is specifically tailored for Cloud Manufacturing Service Composition (CMfg-SC). A comprehensive series of experiments were meticulously executed to assess various facets of the proposed approach, including its effectiveness, efficiency, robustness, adaptability, and scalability in addressing challenges inherent in CMfg-SC. The outcomes of these experiments reveal that, in direct comparison with conventional DQN and Q-Learning algorithms, our proposed PD-DQN algorithm consistently exhibits superior performance. Experiments were conducted by taking production of automotive engine parts such as valve, EGR passage, clutch housing, and oil pan from as an application scenario, in which representative subtasks can be valve, passage, crankcase, gear housing, and oil pan, etc. Different simulations considered several tasks from 10 to 40 subtasks and there are 30 candidate services for each subtask. In all scenarios PD-DQN demonstrated a better convergence and higher rewards compared to DQN, and Q-Learning. This signifies its potential to serve as a more effective solution in the realm of QoS-aware service composition for cloud manufacturing.

- Deep Reinforcement Learning for Semiconductor Production Scheduling [7]

In this research contribution, Reinforcement Learning (RL) employing a Deep Q-Network (DQN) agent was effectively implemented for the domain of production scheduling. The innovative system demonstrated the capability to autonomously generate globally optimal scheduling solutions, eliminating the need for human intervention or any predefined expert knowledge. Notably, the system exhibited a remarkable adaptability, with the capacity to undergo training and adaptation within a matter of hours. The inherent flexibility of the system enables it to dynamically adjust to predefined objectives. Specifically, in the examined case, there was a noteworthy reduction in the share of delayed lots. For instance, in the Technology Classe (TC 1), the percentage of delayed lots decreased from 17% to an impressive 1.3%. Simulations were conducted considering a semiconductor wafer processing. This outcome highlights the system's proficiency in optimizing scheduling objectives and addressing challenges in the production scheduling domain.

- Optimization of global production scheduling with deep reinforcement learning [87]

This paper reports an application of Reinforcement Learning (RL) using the Deep Q-Network (DQN) agent for the domain of production scheduling. The system showcased an autonomous capability to generate scheduling solutions that align with expert benchmarks, all achieved without human intervention or the need for prior expert knowledge. Although the developed algorithm did not surpass existing heuristics, it demonstrated the remarkable achievement of reaching expert-level performance within a mere two days of training. The system exhibited a keen ability to identify non-optimal rules or implementation errors, exemplified by the detection of issues like the introduction of 30% random actions at work center 2. The transparent nature of the system, characterized by a direct connection between the solution and global optimization targets, adds to its appeal. Additionally, the system's quick training and exchange capabilities, accomplished within a matter of hours, further contribute to its effectiveness in the field of production scheduling.

- Optimized Adaptive Scheduling of a Manufacturing Process System with Multi-Skill Workforce and Multiple Machine Types: An Ontology-Based, Multi-Agent Reinforcement Learning Approach [70]

In this paper, the authors introduce an ontology that describes a manufacturing system with multiple stages, machines, and products, all managed by a workforce with diverse skills. They employ a multi-agent system to simulate scheduling and human resource agents, each striving to achieve their respective objectives. The agents collaborate through depth-limited search algorithms. This framework not only simulates real working processes but also takes a systematically cooperative, data-driven approach to adaptively reach scheduling decisions. The results demonstrate a progressive increase in rewards and robust convergence, considering factors such as workers' salaries and the number of options for staff assignments.

7.2.1. Pioneering Contributions to the State of the Art:

The articles conduct an extensive literature review focusing on the applications of Reinforcement Learning (RL) in the domain of machine scheduling problems. They critically analyse key aspects of RL as applied to machine scheduling, highlighting commonly addressed problem types, objectives, and constraints. Additionally, the reviews identifies both shortcomings and promising areas within the existing literature. RL has found application in diverse scheduling challenges within the manufacturing sector, encompassing areas like semiconductor manufacturing systems, cloud manufacturing, reconfigurable manufacturing, robots' applications, and smart factories.

7.2.2. Advantages of Reinforcement Learning in Scheduling:

Reinforcement Learning (RL) boasts several advantages that set it apart from other machine learning methods. Notably, RL excels in addressing intricate scheduling problems featuring multiple objectives and constraints. Its ability to learn from experience and adapt to dynamic environments renders it particularly suitable for scenarios involving dynamic scheduling challenges. Additionally, RL stands out in optimizing scheduling decisions by considering the long-term repercussions of each decision. This stands in contrast to traditional scheduling methods, which may struggle with complex problems and adapting to changing environments.

7.2.3. Areas for improvement and future directions:

Considering the papers that focus on the application of RL in cloud manufacturing the principal areas for improvement are:

- enhancements in adaptability to dynamic production demands, changing task priorities, scalability, real-time responsiveness, machine breakdowns and resources' dependency.
- designing proper reward functions, selecting proper neural networks, and defining action and state space in a reasonable way.
- ensure more efficient management of the data sparsity problem.
- better encoding of jobs
- application of customised RL dispatching rules to different machines

Reinforcement learning (RL) has demonstrated significant potential in addressing scheduling challenges within the manufacturing sector. Nevertheless, several obstacles and constraints must be overcome to advance the application of RL in manufacturing scheduling. The forthcoming directions for RL applications in manufacturing scheduling consider the following aspects:

1. Management of High-Dimensional State and Action Spaces: The curse of dimensionality poses a challenge for RL algorithms when confronted with high-dimensional state and action spaces. To overcome this challenge, prospective research should concentrate on devising more efficient RL algorithms proficient in handling such complex spaces. An illustration of this can be found in the study 'Graph neural networks-based scheduler for production planning problems using reinforcement learning [21] 'where the authors introduced a scheduler for production planning issues utilizing graph neural networks and reinforcement learning. This approach adeptly manages high-dimensional state and action spaces, yielding scheduling decisions near optimality.

2. Reduction in Training Data Requirements: RL algorithms often demand substantial training data and training simulations to attain optimal performance, a demand that may prove impractical in certain manufacturing contexts. To counteract this limitation, future research endeavours should aim to formulate RL algorithms capable of achieving optimal performance with reduced training data and simulations. An example of this lies in the study, 'Inverse Reinforcement Learning Framework for Transferring Task Sequencing Policies from Humans to Robots in Manufacturing Applications [58]', enabling the transfer of task sequencing policies from humans to robots in manufacturing. This approach, grounded in human demonstrations, attains scheduling decisions near optimality with minimized training data.

3. Management of Uncertainty and Stochasticity: RL algorithms may encounter challenges in handling uncertainty and stochasticity inherent in scheduling environments. To address this issue, future research should focus on developing RL algorithms equipped to navigate uncertainty and stochasticity. A case in point is the paper 'Reinforcement learning-based dynamic production-logistics-integrated tasks allocation in smart factories [41]' proposing a reinforcement learning-based dynamic production-logistics-integrated task allocation system in smart factories. This approach effectively manages uncertainties and stochastic elements in scheduling environments, resulting in near-optimal scheduling decisions.

4. Integration of RL with Other Optimization Techniques: Combining RL with other optimization techniques stands as a viable strategy to enhance scheduling decisions. An instance of this integration is evident in 'A multi-objective reinforcement learning approach for resequencing scheduling problems in automotive manufacturing systems [42]'. This approach successfully integrates RL with multi-objective optimization techniques, yielding scheduling decisions approaching optimality.

7.3. Application of Reinforcement Learning for Process control

- Reinforcement learning for process control with application in semiconductor manufacturing [45]

The paper addresses the critical issue of process control in semiconductor manufacturing, aiming to minimize process variation by obtaining optimal control actions based on historical offline data and real-time system output. In contrast to traditional control methods that rely on linear process models, the work introduces RL-based controllers, which are more versatile and not confined to specific process models. The proposed RL-based controllers are developed with consideration of domain knowledge availability for approximating process models. Two RL-based control algorithms are presented, and their theoretical properties are discussed based on widely accepted linear process models. Simulations in two scenarios demonstrate that RL-based controllers are not only superior or at least comparable to traditional controllers, such as the GHR controller, in linear cases, but also exhibit promising potential for handling more complex, non-linear scenarios.

- A reinforcement learning approach for process parameter optimization in AM [14]

The study addresses a process parameter optimization challenge using an on-the-fly model-free Q-learning-based reinforcement learning (RL) approach. The optimization focuses on determining laser power (P) and scan velocity (v) combinations to maintain a steady-state melt pool depth (δ) of 1 mm in selective laser melting (SLM) of SS316L material. The RL framework is set up with the laser as the agent and the L-DED (laser-directed energy deposition) system emulated via an Eagar–Tsai function as the environment. The optimal $P - v$ combination predicted by the algorithm is 888.9 W - 566.7 mm/min, resulting in a melt pool depth within 50 μm of the experimental observation. Comparison with an experimentally derived process map shows a deviation within 50 μm. The study also analyzes the effects of hyperparameters, such as discretization, exploration–exploitation parameter, discount factor, learning rate, and number of episodes, on the Q-learning process. This methodology provides a versatile solution for process parameter optimization in scenarios with limited system information or data availability, applicable to various additive manufacturing or advanced manufacturing systems beyond L-DED.

- Hybrid feedback and reinforcement learning-based control of machine cycle time for a multi-stage production system [43]

This paper introduces a novel control method designed for multi-stage production systems, aiming to dynamically adjust the cycle time of individual machines for enhanced overall system efficiency. The proposed method combines a distributed feedback control scheme with a Reinforcement Learning (RL) control scheme utilizing an extended actor-critic algorithm (A2C).The feedback control aspect determines whether a machine should be turned on or off based on real-time system status, while the RL control scheme decides how to adjust a machine's cycle time when it is in operation. The extended actor-critic RL algorithm incorporates a model-based path, enhancing learning performance compared to standard model-free RL approaches.Numerical case studies were conducted to demonstrate the effectiveness of the proposed method. Results indicate significant improvements in overall profits and energy savings compared to other methods. Specifically, the hybrid control scheme outperforms the standard A2C by 29.62% in terms of total cost, while the extended A2C algorithm improves system performance by 23.4% compared to the standard A2C.

- Graph neural network and multi-agent reinforcement learning for machine-process-system integrated control to optimize production yield [25]

This paper introduces an integrated control framework aimed at optimizing production yield by incorporating various levels of a manufacturing system, encompassing system, process, and machine levels. The manufacturing system is conceptualized as a graph, where machines represent nodes and material flows act as links. The graph model offers flexibility and accommodates real-time information across different levels by dynamically updating node features.To enhance decision-making accuracy, Recursive Bayesian Estimation (RBE) is employed to refine tool state observations obtained from sensors and machine learning models. The refined tool state estimations are integrated into the graph node features. Employing the graph model, Graph Neural Network (GNN) processes node features, generating embeddings that capture both local and global information. For integrated control, each machine node functions as a distributed agent in a Multi-Agent Reinforcement Learning (MARL) setting. The agent conditions its policy on the node embedding from GNN. State-of-the-art GNN and MARL algorithms, specifically Graph Attention Network (GAT) and Value Decomposition Actor Critic (VDAC), are implemented to train learnable parameters in the GNN-MARL networks, facilitating the learning of an optimal multi-agent policy. Extensive numerical experiments and analysis validate the effectiveness of the proposed integrated control framework.

- Dynamic Control of a Fiber Manufacturing Process Using Deep Reinforcement Learning [32]

This article introduces a model-free deep reinforcement learning (DRL) approach for the control of a fiber drawing system. The custom DRL-based control system proactively regulates fiber diameter, ensuring a desired, constant, or variable diameter trajectory along the fiber length. The approach does not rely on physical models of the system. The system was trained and tested on a compact fiber drawing system characterized by nonlinear delayed dynamics and stochastic behaviors. When subjected to a reference trajectory with random step changes, the DRL controller, after 1 hour of training, exhibited a root-mean-squared error (RMSE) comparable to an optimized Proportional-Integral (PI) controller. After 3 hours of training, it achieved performance like that of a quadratic dynamic matrix controller (QDMC). In a step response, the PI feedback controller showed a 3.5-second time lag, while the DRL controller exhibited less than a second of time lag. Controller performance tests on trajectories not used in the training process were conducted. For a sine sweep reference trajectory, the DRL controller maintained an RMSE under 40 µm up to a frequency of 45 mHz, compared to 25 mHz for QDMC.

- Explainable Deep Reinforcement Learning For Production Control of job shop manufacturing system [36]

In response to the increasing complexity of material flows due to a rising number of variants and smaller batch sizes in manufacturing, this publication introduces an approach to enhance production planning and control (PPC). The focus is on creating a more functional and user-friendly PPC system by integrating multiagent reinforcement learning (MARL), a successful approach in ML-based production control, along with methods for explaining decisions made by reinforcement learning (RL) algorithms. MARL enables short reaction times and high decision quality. The developed MARL system is then combined with explainable Artificial Intelligence (XAI) methods to enhance user trust. The use case

results demonstrate that the developed system can outperform rule-based controls commonly used in industry while providing explainable decisions. To assess the system's performance, two conventional methods, First in - First out (FIFO) and shortest set-up time next (SSTN), are used for comparison. In a simulation of 52 episodes, MARL exhibits an average total reward that is 22% higher compared to FIFO and 15% higher compared to SSTN.

- Multi-objective optimization of the textile manufacturing process using deep-Q-network based multi-agent reinforcement learning [22]

In this study, they propose a multi-agent reinforcement learning (MARL) methodology designed to address the escalating complexities of multi-objective optimization problems within the textile manufacturing process. The optimization of textile process solutions with multiple objectives is conceptualized as a stochastic Markov game. Multiple intelligent agents, leveraging deep Q-networks (DQN), are developed to attain correlated equilibrium optimal solutions for the optimization process. The stochastic Markov game is characterized by neither complete cooperation nor full competition. To navigate this balance, agents employ a utilitarian selection mechanism that maximizes the sum of all agents' rewards, following an increasing ε-greedy policy in each state to avoid disruption caused by multiple equilibria. Case study results demonstrate that the proposed MARL system can achieve optimal solutions for the textile ozonation process and enzyme washing process, outperforming traditional approaches. Notably, the agents are trained to optimize the ozonation process solution efficiently, achieving the desired color on treated fabrics. The relatively shorter computation time and higher performance of the MARL system can be attributed to the parallel operation of multiple agents and their ability to share experiences during the process. Conversely, metaheuristic algorithms such as MOPSO and NSGA-2 may struggle with smaller datasets and impractically long iteration times. Additionally, their effectiveness diminishes when dealing with higher-dimensional multi-objective optimization problems, making the proposed MARL system a promising alternative.

- Modular production control using deep reinforcement learning: proximal policy optimization [54]

This article explores the application of Deep Reinforcement Learning (DRL), specifically the Proximal Policy Optimization (PPO) method, to the challenge of Model Predictive Control (MPC) in the context of modular production systems within the automotive industry. The complexity and diversity of these systems make them challenging to control, and RL algorithms offer a powerful and versatile approach.The Proximal Policy Optimization (PPO) agent effectively coordinates the production system, providing a reliable solution to the Model Predictive Control (MPC) problem. The utilization of parallel environments and the repetitive use of collected trajectories for updating the policy contribute to a stable and robust learning environment for the specific example discussed. The intention is to demonstrate this effectiveness in addressing higher complexity problems in future applications.

- Reinforcement Learning for Statistical Process Control in Manufacturing [80]

The paper introduces the innovative concept of integrating Reinforcement Learning (RL) into Statistical Process Control (SPC) within the manufacturing domain. The necessary elements for

incorporating RL into SPC, including states, actions, and rewards, were defined. The Q-Table method was employed for stable and predictable results, requiring quantization of time series values and Quality Control Charts (QCC) into stripes. The state vector was formed by recent stripes of production trend values and selected production actions. Manufacturing interventions to keep measured production values within tolerance range constituted the RL action list. The manufacturing goal was to minimize production unit cost while maintaining a high ratio of good products. The RL reward solution included the cost of applied actions and the cost of failure products. A dynamic Q-Table technique was introduced, allocating memory as needed, which is practical for real-world applications. Two additional concepts, Reusing Window (RW) and Measurement Window (MW), were introduced to address the cost of a measurement value and the precise evaluation requirement in the manufacturing SPC environment. The paper also describes novel RL extensions, such as epsilon self-control of exploration and exploitation, and the optimization of training meta-data. Performance comparison involved analyzing the distribution of selected action frequencies, unit prize, and the rate of good products. The ratios of selected "No action" type actions were crucial KPIs. In the best setups, the proposed concept resulted in 10–30% fewer production intervention actions than in the manufacturing shopfloor, demonstrating promising applicability through industrial testing and validation.

- Digital Twin and Reinforcement Learning-Based Resilient Production Control for Micro Smart Factory [64]

To enhance the operational efficiency of Manufacturing Service Factories (MSF) within Cyber-Physical Production Systems (CPPS), this paper introduces resilient production control methods based on Digital Twin (DT) and Reinforcement Learning (RL). The primary objective is to facilitate the learning of an RL policy network, replacing the conventional dispatching rule in the post-processing station of MSF. The proposed method is designed with careful consideration of technical requirements, taking into account the restructuring nature of Manufacturing Management Systems (MMS) and the need for robustness in the system. The inherent complexity introduced by the Make-to-Order (MTO) production environment in personalized production within MSF is also addressed. The technical functionalities essential for CPPS in MSF play a crucial role in achieving system resilience.In the implementation phase, the method relies on the coordination between the DT application and the policy network construction module. The DT application performs tasks such as creating, synchronizing, and utilizing the DT to offer simulation support as part of its technical functionalities within CPPS. The DT simulation, generating virtual event logs, aids in the learning process of the RL policy network. Conversely, the policy network construction module employs the dueling network technique to learn and apply the RL policy network. The learning process is guided by actions, states, and rewards derived from virtual event logs.Moreover, the proposed method emphasizes the need for synchronization of dynamic information, including production progress volume, Work In Progress (WIP), machine status, and changes in the operational situation, within the Digital Twin The iterative creation process of the DT application consistently mirrors the RL policy network, while the utilization process of the DT application continually evaluates the RL policy network. This integrated approach aims to optimize the CPPS in MSF, contributing to improved efficiency and resilience in manufacturing operations.

- A Deep Reinforcement Learning approach for the throughput control of a FlowShop production system [53]

In this study, a novel approach based on Deep Q-Network (DQN) was proposed to control a flow shop. The primary objective was to address inefficiencies by controlling Work in Process (WIP) and maintaining a stable Throughput (TH). This method combines Reinforcement Learning (RL) and Deep Neural Networks (DNN) to model the state and action space in a unique way, specifically tailored for controlling WIP and TH in a 5-machine flow shop. The state was represented by considering job completion times on each machine and the deviation between the current TH on a machine and the TH-target for the five workstations. The action space aimed to assess the WIP amount in the system to control the production line's TH concerning the TH-target. The dataset was generated through simulations, and the results of the proposed approach showed promise, especially given the high variability of the experimental scenario. The study emphasized that smaller DNN structures achieved better performance, enabling faster training without sacrificing the generality of the learning process. The practical application of this methodology in Industry 4.0 involves submitting data from the production line to a central controller for decision-making in response to disruptions.The presented approach outperformed the Practical Worst Case (PWC) that was used as a benchmark to compare the performances of the introduced DQN.

- Reinforcement Learning Based Production Control of Semi-automated Manufacturing Systems [60]

In this study, a combination of reinforcement learning and digital twin methods is employed to devise a production control logic within a semi-automated production system following the chaku-chaku principle. The reinforcement learning method is embedded into the digital twin to autonomously learn an optimized production control logic for task distribution among different workers on the production line. By analyzing the impact of various reward shaping and hyper-parameter optimization strategies on the quality and stability of results, a well-configured policy-based algorithm is shown to efficiently manage workers and derive an optimal production control logic. The algorithm enhances productivity and ensures stable task assignments, facilitating a seamless transition to daily operations. Validation is performed in the digital twin of a real assembly line of an automotive supplier. The results suggest a novel approach to optimize production control by focusing on workers' routines and leveraging artificial intelligence with a comprehensive overview of the entire production system. The methodology is a collaborative effort between the wbk Institute for Production Technology at the Karlsruhe Institute of Technology (KIT) and the central department Connected Manufacturing of the Bosch Powertrain Solutions division. The implementation and testing were conducted in the simulation model of a real-world production system for car engine components organized in a semi-automated assembly cell based on the Chaku-Chaku principle. The simulation demonstrates that while the number of produced pieces per episode is initially similar, the RL agent significantly improves the output, showcasing an 8% improvement in the number of produced pieces. Additionally, the study reveals that the number of workers on the production line does not impact the learning capacity of the agent, and the number of produced parts per episode continues to increase throughout the learning process..

- Model-free Adaptive Optimal Control of Episodic Fixed-horizon Manufacturing Processes Using Reinforcement Learning [15]

The study demonstrates the applicability of reinforcement learning (RL)-based methods for adaptive optimal control in episodic fixed-horizon manufacturing processes that exhibit varying process conditions. An algorithm based on model-free Q-learning has been introduced, allowing adaptability to changing process conditions by dynamically modifying the Q-function through learning. The proposed algorithm is designed for a specific class of episodic fixed-horizon manufacturing processes. The application of this approach has been exemplified and assessed in the context of optimal control of the blank holder force in a deep drawing process. The primary objective of optimal control was to optimize internal stresses, wall thickness, and material efficiency for the resulting workpiece. The evaluation of the approach involved simulating deep drawing processes using Finite Element Method (FEM). Experimental processes were conducted in an automated virtual laboratory environment, introducing stochastic variations in process conditions and measurement noise.

- A Model-Based Reinforcement Learning and Correction Framework for Process Control of Robotic Wire Arc Additive Manufacturing [2]

"This paper introduces an integrated model-based reinforcement learning-correction framework for in-situ Machine Learning for Metal-Based Additive Manufacturing (MLMB) processes, specifically focusing on robotic Wire Arc Additive Manufacturing (WAAM). The study was conducted on a robotic WAAM system developed at Singapore University of Technology and Design (SUTD), involving experiments with two different metals, namely bronze (ERCuNiAl) and stainless steel (ER316LSi). The experimental results reveal that the print outputs obtained through the proposed learning framework exhibit superior surface finish and are closer to the desired near-net shape. This demonstrates the feasibility and effectiveness of the formulated learning architecture for in-situ process learning and control in the context of robotic WAAM. To further assess the applicability of the developed algorithm, a quantitative comparison of the layer's surface uniformity was performed. The standard deviation (STD) of the surface height for each printed layer was calculated from the layer's surface scan output. The results indicated that the standard deviation of the layer's surface height, when using the recommended single-bead parameters, exhibited an increasing trend as the print height progressed vertically for both materials, suggesting an accumulation of error. The prints implementing the proposed learning framework initially showed a larger surface standard deviation, attributed to the initial learning process where the system explores and evaluates the influence of different manufacturing parameters on MLMB print behavior. As the learning progresses, the system gains a better understanding of the manufacturing process, enabling it to select optimal parameters to correct previous errors and achieve the desired outcome. This results in a more bounded and lower standard deviation, producing a closer-to-net shape output. The encouraging outcomes of this study suggest the potential for cost-effective MLMB process learning, an aspect that has been relatively underexplored due to the high experimental overhead cost and the complexity of modeling involved. For more complex prints, such as the twist lock pin, the learning process demonstrates the system's adaptability and capability to optimize manufacturing parameters, ultimately improving the quality of the output.

7.3.1. Pioneering Contributions to the State of the Art:

These articles represent pioneering advancements in the state of the art of Reinforcement Learning (RL) within the domain of process control applications. They introduce innovative methodologies capable of effectively addressing challenges inherent in complex and dynamic environments, high-dimensional state spaces, sparse rewards, and non-linear and non-convex optimization problems. Notably, the effectiveness of these proposed methods has been substantiated through comprehensive experiments conducted on both real-world and simulated manufacturing processes. These findings collectively contribute to the enhancement of RL techniques in the realm of process control, marking significant progress in the field.

7.3.2. Advantages of Reinforcement Learning in Process Control:

Process control in manufacturing stands as a pivotal element in modern manufacturing systems, striving to ensure optimal efficiency and product quality. Traditional process control methods heavily rely on intricate mathematical models, detailing the correlation between process inputs and outputs. These models often fall short in encapsulating the intricate dynamics of contemporary manufacturing systems. Reinforcement Learning (RL) emerges as a transformative alternative to traditional process control approaches, offering a multitude of advantages. Firstly, RL operates as a model-free methodology, eliminating the necessity for a predetermined mathematical model of the manufacturing process. Instead, the RL agent dynamically learns to control the process through direct interaction with the environment, fostering adaptability to changes in the manufacturing process. Secondly, RL exhibits prowess in handling complex and nonlinear manufacturing processes. While traditional process control methods are typically confined to linear systems, RL showcases the capability to manage nonlinear systems, learning to control intricate processes with multiple variables. Thirdly, RL introduces the capacity to optimize the manufacturing process dynamically over time. In contrast, traditional process control methods often adopt static approaches that struggle to adapt to changes in the manufacturing process. RL's adaptive nature enables continuous optimization and adjustment to changes in the process. Fourthly, RL excels in managing multiple objectives concurrently, a common scenario in diverse manufacturing processes. By learning to optimize multiple objectives simultaneously, RL showcases versatility in addressing the multifaceted goals of manufacturing. Lastly, RL adeptly navigates uncertainty and variability inherent in the manufacturing process. Given the frequent presence of variability and uncertainty, RL's adaptability shines as it learns to navigate and respond to changes and uncertainties in the manufacturing environment. These attributes collectively position RL as a potent and flexible tool in the realm of manufacturing process control, offering innovative solutions to the challenges posed by contemporary manufacturing dynamics.

7.3.3. Areas for improvement and future directions:

Considering the papers that focus on the application of RL in manufacturing process control the principal areas for improvement are:

- practical implementation of the acquired control logic in real-world scenarios. This is imperative because exerting control over workers at a granularity of seconds proves to be infeasible in real applications.

- enhancing the robustness of the agent becomes crucial, especially when faced with alterations in the production system layouts.
- to validate its versatility, the agent needs training in various environments.
- Integrate job routing and deviation management.
- scaling the problem to real-world production cases (it was no feasible in all the works)
- incorporating multiple tasks from different levels into the distributed agents to achieve integration in control functions.,

Reinforcement Learning (RL) is proving to be a promising approach in the field of manufacturing process control. To advance the application of RL in this domain, future research should focus on several crucial aspects. One key area is the development of more efficient RL algorithms tailored for large-scale manufacturing systems. Additionally, exploring hybrid machine learning techniques, such as the integration of RL with neural networks, could enhance the accuracy and efficiency of process control. Another significant avenue for research is the creation of RL-based control systems capable of handling multiple objectives simultaneously (quality, cost, energy management, etc). Many manufacturing processes involve diverse objectives that require optimization, and RL has shown promise in effectively managing this complexity. Furthermore, addressing the challenges associated with uncertainty and variability in manufacturing processes is essential. RL has demonstrated an ability to adapt to such dynamic conditions, and future research should concentrate on refining RL-based control systems to handle these challenges seamlessly. Lastly, the translation of RL-based control systems from theoretical frameworks to practical implementation in real-world manufacturing settings poses challenges. Future research efforts should be dedicated to overcoming these challenges, ensuring that RL-based control systems can smoothly integrate into and benefit real-world manufacturing environments.

7.4. Applications of Reinforcement Learning in Autonomous Manufacturing

- Reinforcement Learning Enabled Autonomous Manufacturing Using Transfer Learning and Probabilistic Reward Modelling [50]

The paper introduces a reinforcement learning (RL)-enabled autonomous manufacturing system (AMS) designed to autonomously fabricate complex geometry artifacts with desired performance characteristics. Addressing the sample inefficiency issue of traditional RL algorithms in real-world manufacturing decision-making, the approach leverages a first-principles-based source task for training, transfers effective representations from the acquired knowledge, and utilizes these representations to interact with the physical system and learn a probabilistic model of the target reward function. The method is applied to a custom physical AMS machine capable of autonomously manufacturing phononic crystals, demonstrating the effectiveness of the approach in modeling the target reward function with a small number of artifacts, as low as 25, and finding artifacts with high reward. This is a significant improvement over traditional methods that often require manual design and extensive empirical iterations on the order of hundreds.

- Reinforcement learning and optimization-based path planning for thin-walled structures in wire arc additive manufacturing [65]

In this study, a successful demonstration of planning the deposition path and determining process parameters for thin-walled structures was achieved. The deposition path planning utilized the reinforcement learning approach Proximal Policy Optimization (PPO), while an optimization technique was employed to determine process parameters such as welding speed and wire feed rate. The framework developed for this path and process parameters planning was named RLPlanner. The input models were parameterized in Cartesian coordinates, providing a fast and straightforward approach. Additionally, the input model was parameterized layer by layer based on the height of the weld bead in the previous layer, allowing the algorithm to operate in 2D space instead of 3D. This resulted in the use of AI architectures with fewer trainable parameters, enhancing memory and time efficiency. The reinforcement learning agent received input not for the entire parameterized layer but only for its surroundings, defined as the field of view, contributing to a memory and time-efficient solution. RLPlanner demonstrated the capability to adjust welding speed and wire feed speed between layers, providing adaptability to varying input 3D geometries. Furthermore, when the input model consisted of separate parts, they were localized and processed individually by the algorithm. Notably, the presented solution for path planning required no human intervention, avoided path templates, and was easily implementable.

- Optimisation of manufacturing process parameters for variable component geometries using reinforcement learning [103]

This study introduces a Reinforcement Learning (RL)-based approach for estimating optimal manufacturing parameters in the context of variable component geometries. The specific focus is on positioning pressure pads to optimize material draw-in during fabric forming, particularly for cuboid boxes. Unlike classical surrogate-based optimization (SBO), the proposed approach trains a function (P) that takes the component geometry as input and directly estimates optimal process parameters as output. The training process occurs in an FE-simulation environment. The trained network demonstrates the ability to provide meaningful parameter estimations even for new geometries not included in the training set, showcasing its capacity to extract reusable information from generic process samples and apply it successfully to novel, non-generic components. The approach, which involves reusing data rather than resampling, is considered a promising avenue for lean part and process development.

- Collaborative Clustering Parallel Reinforcement Learning for Edge-Cloud Digital Twins Manufacturing System [16]

In this study, a pioneering deployment and execution pattern for Digital Twins (DTs) has been introduced, showcasing reduced interaction delay and improved analysis delay convergence rates. The foundation of this advancement lies in the establishment of a collaborative DTs application deployment architecture that synergizes cloud and edge computing. Within this architecture, deterministic and uncertainty application adaptive strategies have been incorporated. To address the challenges posed by adaptive scenarios, they have presented distributed Closed-Loop Proportional-Integral-Quantized-Learning (CCPQL) and prediction-based CCPQL algorithms. Simulation results illustrate the superior performance of the proposed algorithm in comparison to conventional

methods (tradition QL, Sarsa, state-of-art RCMP). This promising outcome signifies potential efficiency enhancements in real manufacturing applications.

- Towards Self-X cognitive manufacturing network: An industrial knowledge graph-based multi-agent reinforcement learning approach [99]

The readiness of 'Self-X' levels, such as self-configuration, self-optimization, and self-adjust/adaptive/healing, is still in its early stages. This work aims to pave the way for these advancements by introducing a stepwise approach using an industrial knowledge graph (IKG)-based multi-agent reinforcement learning (MARL) method to achieve a Self-X cognitive manufacturing network. The proposed methodology involves the formulation of an IKG based on empirical knowledge and recognized patterns in the manufacturing process. This is achieved by leveraging extensive human-generated and machine-sensed multimodal data. Subsequently, a graph neural network-based embedding algorithm is applied, drawing upon a comprehensive understanding of the established IKG. This step enables semantic-based self-configurable solution searching and task decomposition. Furthermore, a MARL-enabled decentralized system is introduced to self-optimize the manufacturing process. This system works in tandem with the IKG to contribute to the realization of a Self-X cognitive manufacturing network. To validate the feasibility of this approach, an illustrative example of a multi-robot reaching task is conducted. In summary, the proposed method offers a structured approach to harnessing industrial knowledge graphs and MARL for achieving self-configurable and self-optimizing capabilities in a cognitive manufacturing network. The illustrative example demonstrates the potential feasibility and effectiveness of the proposed approach.

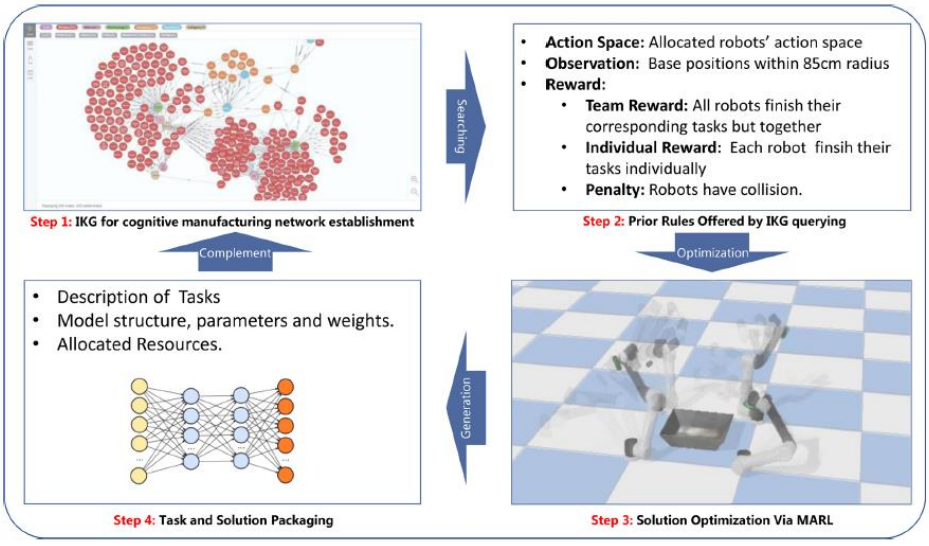

Figure 10. Core steps of the simulated example.

- Fault-Tolerant Control of Programmable Logic Controller- Based Production Systems with Deep Reinforcement Learning [104]

This article presents a proof of concept for the application of deep reinforcement learning (DRL) to automatically restart PLC-based automated production systems (aPS) during fault recovery. The focus is on aPS with multiple end-effectors actuated in one or two axes, particularly applicable to systems

for assembly and logistics tasks. To address challenges such as an expanding action space and the absence of a global coordinate system for workpiece tracking, the authors propose a hierarchical Multi-Agent System (MAS) with a separate coordinate predictor for each agent. Each module of the aPS, representing an independent subunit with specific functionality, is treated as a single agent in the MAS. The evaluation of the concept involves simulating a laboratory demonstrator composed of actuators like pneumatic cylinders and conveyors, commonly found in real-world aPS. The results show that the DQN (Deep Q-Network) algorithm can effectively learn the control of widely used modules such as separators and conveyor systems. For more complex modules like cranes, DQN's exploration needs some support with a slightly shaped reward function. On the other hand, PPO (Proximal Policy Optimization) can only learn basic modules due to its limited exploration capabilities compared to DQN. The trained hierarchical MAS successfully enables the restart of the laboratory demonstrator from various states, even those not part of the standard control trajectory. The use of DRL for the restart of aPS is demonstrated to be feasible, and the hierarchical approach allows scalability to aPS with numerous modules without significantly increasing the action space of an individual agent. However, the size of supported modules is limited by the number of actuators and their complexity that can be explored.

- Reinforcement learning for facilitating human-robot-interaction in manufacturing [59]

This study addresses the challenge of improving the adaptability of robotic operators to variations in human task performance within contemporary manufacturing processes. The work introduces a methodology for effective system modeling and the development of a reinforcement learning agent capable of autonomous decision-making. This agent enhances the adaptability of robotic operators by allowing them to adjust their behavior based on observed information from the environment and human colleagues. The study contributes to theoretical knowledge on implementing learning methods for robotic control and leveraging these capabilities to enhance human-robot interactions. The evaluation, conducted in a generalized simulation model parameterized for human performance variation, demonstrates that the reinforcement agent effectively learns to adjust its behavior based on observed information and optimize task demands.

7.4.1. Pioneering Contributions to the State of the Art:

The listed articles represent main applications and the state of the art if RL in autonomous manufacturing. It is applied in different scenarios as additive manufacturing, self-adaptive manufacturing networks and in collaboration with digital twins and robots setting. One of the pioneers is the article ‘Optimisation of manufacturing process parameters for variable component geometries using reinforcement learning [103]’ which demonstrates how RL optimizes process parameters for varying component shapes. It ensures consistent quality while accommodating design variations. Another relevant contributor is the work ‘Reinforcement learning and optimization-based path planning for thin-walled structures in wire arc additive manufacturing [65]’ where RL optimizes path planning in additive manufacturing, leading to improved material deposition, reduced defects, and enhanced structural integrity. These are just a couple of examples of how RL can solve autonomous manufacturing challenges that enable flexibility and process control to be able to face

new manufacturing questions that are arising from rapid transformations due to globalization, digitalization, and personalization.

7.4.2. Advantages of Reinforcement Learning in autonomous manufacturing:

Considering the listed papers, advantages of RL in autonomous manufacturing, which reveals its transformative potential, are such that the algorithm effectively manages order dispatching in time-constrained job shops, outperforming traditional benchmarks. In the realm of additive manufacturing, RL optimizes path planning, leading to improved material deposition and enhanced structural integrity. The adaptability of RL to complex geometries and real-time adjustments proves to be a significant advantage and it prowess in optimizing process parameters for varying component shapes, ensuring consistent quality across diverse designs and efficiency in edge-cloud computing integration, offering scalability and fault tolerance. Additionally integrating RL with knowledge graphs enables self-adaptive manufacturing networks that adapt to changing environments and facilitate knowledge sharing. RL also enhances fault tolerance by learning robust control policies. Finally, RL proves beneficial in enhancing collaboration between humans and robots, adapting to dynamic environments, and ensuring safety. In summary, RL emerges as a versatile and powerful tool, offering adaptability, efficiency, and fault tolerance, making significant strides in optimizing complex manufacturing processes within autonomous systems.

7.4.3. Areas for improvement and future directions:

Considering the papers that focus on the application of RL in manufacturing PPC, the principal areas for improvement are:

- enhancing collaboration between humans and robots using reinforcement learning (RL) necessitates the consideration of safety, interpretability, and the development of intuitive interfaces.
- seamless integration of systems, the creation of user-friendly interfaces, and the implementation of adaptive learning approaches.
- the algorithm should be evaluated on the physical laboratory demonstrator.

The integration of reinforcement learning (RL) into autonomous manufacturing represents a promising era, but not without its share of challenges. From the sensitivity of RL models to environmental variations, as highlighted in "Reinforcement Learning Enabled Autonomous Manufacturing Using Transfer Learning and Probabilistic Reward Modelling [50]," to the complex coordination among multiple agents in manufacturing, as discussed in "Collaborative Clustering Parallel Reinforcement Learning for Edge-Cloud Digital Twins Manufacturing System [16]" each challenge presents a unique set of obstacles.
Future directions start from developing robust RL algorithms to handle uncertainties and sensor noise to enhancing sample efficiency through meta-learning and imitation techniques. Emerging trends such as interpretable RL models, multi-agent systems, and adaptive learning mechanisms, emphasizing their crucial roles in steering RL towards an impactful future in shaping the autonomous manufacturing landscape.

7.5. Applications of Reinforcement Learning for Maintenance Strategies and Quality

- Post-prognostics demand management, production, spare parts and maintenance planning for a single-machine system using Reinforcement Learning [88]

The study focuses on the comprehensive planning of joint spare parts sourcing, inventory management, production, and maintenance for a single machine across multiple periods to meet customer demands. The primary objective is to maximize production revenue while minimizing costs. A data-driven post-prognostics Reinforcement Learning (RL) model was developed to enhance and automate decision-making in Production Planning and Control (PPC). Unlike previous research, this RL model integrates maintenance decisions and extends to include PPC decisions, addressing demand management, spare parts sourcing, and production planning simultaneously. The RL model incorporates a data-driven prognostics model, utilizing a regressive random forest algorithm, to forecast the next health state of the production machine with exceptional performance. The constructed problem environment includes action and state spaces, state transitions, and reward signals to create a realistic and practical scenario. Three different RL algorithms (DQL, PPO, A2C) were developed and evaluated, with PPO demonstrating superior performance compared to other RL models and traditional Reliability Centered Maintenance (RM) and Preventive Maintenance (PM) strategies. Through sensitivity analysis, the robustness of the PPO algorithm was demonstrated under increased levels of noise and different cost scenarios. This study represents a significant advancement by integrating PPC decisions with maintenance decisions and leveraging data-driven prognostics within an RL framework, ultimately providing a more holistic and effective approach to decision-making in manufacturing.

- Multi-objective reinforcement learning-based framework for solving selective maintenance problems in reconfigurable cyber-physical manufacturing systems [4]

This paper introduces a robust model for a multistate, multi-component Reconfigurable Cyber-Physical Manufacturing System (RCPMS) that considers imperfect repairs. The model incorporates layout configuration selection and addresses uncertainties stemming from imperfect observations of components' health status. The objectives of the model are to maximize expected reliability, minimize variance and maintenance cost, all under time and production capacity constraints.To solve the resulting multi-objective and combinatorial optimization problem, the paper proposes a novel deep reinforcement learning framework. This framework is designed to handle the complexity of the problem and incorporates decision values to enhance the scalarization process. Decision values allow the adjustment of priorities for specific objectives after the learning process while maintaining overall performance. Additionally, the framework is combined with Analytical Hierarchy Process (AHP) to dynamically update static decision-maker priorities based on the actual learning context.The proposed model and Multi-Objective Reinforcement Learning (MORL) framework are extensively evaluated through various experiments, demonstrating their performance and robustness in challenging scenarios. The impact of AHP is analyzed by comparing results obtained with the MORL framework using static priorities. The findings emphasize the effectiveness and adaptability of the proposed approach in addressing complex multi-objective optimization problems in the context of Reconfigurable Cyber-Physical Manufacturing Systems.

- Joint optimization of maintenance and quality inspection for manufacturing networks based on deep reinforcement learning [93]

This study delves into the joint optimization of maintenance and quality inspection in manufacturing networks using DRL, considering interactions between machine reliability and Work-in-Progress (WIP) quality. The research begins by proposing mathematical models that capture the nonlinear, high-dimensional, and dynamic nature of manufacturing network environments. These models offer a robust state transition representation for controlling manufacturing networks. Subsequently, an efficient DRL model is developed to handle the joint control of reliability and quality in manufacturing networks, accommodating mixed discrete-continuous states and actions simultaneously. The DRL model is validated through contrast training with a generic algorithm (GA), highlighting its superior adaptability to dynamic and diverse manufacturing scenarios compared to GA. Experimental results demonstrate that the proposed models effectively balance the trade-off between economic profit and operational risk in manufacturing networks.

- Dynamic Maintenance for a Large Scale Identical Parallel Manufacturing Systems Using Reinforcement Learning [52]

The authors have presented a reinforcement learning (RL)-based framework for maintenance decision-making, aiming to minimize costs. The study focuses on a parallel multi-unit system subject to independent random failures. Each component within the system can exist in one of three states: healthy, unhealthy, or failed. The researchers applied a Q-learning algorithm to derive the optimal maintenance policy for the system. The effectiveness of the proposed model is evaluated through a numerical example and sensitivity analysis. Furthermore, the proposed model is compared with two alternative policies. The results demonstrate significant reductions in total maintenance costs, with the proposed algorithm achieving the lowest total cost compared to the alternatives (39963148 vs. 39981000 and 39992000 for the compared policies).

- A Reinforcement Learning Algorithm for Optimal Dynamic Policies of Joint Condition-based Maintenance and Condition-based Production [19]

This paper focuses on developing a joint optimal maintenance and production policy for a specific type of production system with adjustable production rates. The rate of system deterioration is directly linked to the production rate. The deterioration can be controlled through maintenance actions (maintenance policy) and adjusting the production rate (production policy). The problem is modeled as a Markov decision process (MDP), and a reinforcement learning algorithm, specifically Q-learning, is employed to determine optimal actions based on the system's state. The goal is to minimize expected costs over a finite planning horizon. The algorithm's hyperparameters are tuned using a value-iteration algorithm of dynamic programming. The Q-learning algorithm performs well in minimizing expected costs. For instance, in state (4; ns), the optimal value function is 127, and the optimal action is to schedule maintenance and set the production rate to the maximum level (1).

- Reinforcement learning-based defect mitigation for quality assurance of additive manufacturing [11]

The main challenge in the additive manufacturing (AM) industry lies in ensuring quality assurance due to the potential time-varying processing conditions during the AM process. The emergence of new defects during printing, which cannot be addressed by offline analysis tools focused on existing defects, adds complexity to this challenge. This paper responds to this issue by introducing online learning-based methods to tackle new defects during printing, particularly in the context of fabricating a small number of customized products in AM.The proposed method is based on model-free Reinforcement Learning (RL) and is named Continual G-learning. This approach aims to minimize the number of samples needed for defect mitigation in the AM process. Continual G-learning leverages prior knowledge from various sources to enhance its performance. Offline knowledge is gathered from literature, while online knowledge is acquired during the printing process. The method introduces a novel algorithm for learning optimal defect mitigation strategies, demonstrating superior performance when utilizing both knowledge sources.The effectiveness of the proposed method is validated through numerical and real-world case studies conducted on a fused filament fabrication (FFF) platform. The results highlight the efficacy of Continual G-learning in mitigating defects during AM, showcasing its potential for improving the quality assurance process in additive manufacturing.

- Deep multi-agent reinforcement learning for multi-level preventive maintenance in manufacturing systems [75]

To devise cost-efficient preventive maintenance (PM) policies for a serial production line featuring multiple levels of PM actions, this study adopts a novel multi-agent modeling approach. Each machine is modeled as a cooperative agent to facilitate adaptive learning. The reward function is constructed based on the evaluation of system-level production loss. An adaptive learning framework, utilizing the value-decomposition multi-agent actor–critic (MARL) algorithm, is employed to derive effective PM policies.In simulation studies, the proposed framework proves its efficacy by outperforming other baselines across a comprehensive set of metrics. Notably, centralized RL-based methods struggle to converge to stable policies. The authors conduct two numerical experiments to emphasize the importance of modeling PM decision-making as multi-agent problems, contrasting it with a DQN single-agent approach used in their prior work. In the 6-machine-5-buffer experiment, the DQN policy faces convergence issues, while the MARL policy achieves the best profit among all baselines. In the 10-machine-9-buffer experiment, the DQN method encounters implementation inefficiencies due to the growing size of its replay buffer and action space. In contrast, MARL does not suffer from this issue and consistently reports the best average profit among all policies.

- A Novel Reinforcement Learning-based Unsupervised Fault Detection for Industrial Manufacturing Systems [1]

In real-world scenarios, the lack of knowledge about relevant features reflecting actual machine conditions poses challenges in addressing fault diagnosis problems. Machine learning (ML) approaches often require customized models and ad-hoc feature extractions for each case study. Additionally, the early substitution of key mechanical components for preventing breakdowns presents difficulties in collecting sizable datasets to train fault detection (FD) systems. To overcome these challenges, this paper introduces a novel unsupervised FD method based on a double deep-Q network (DDQN) with prioritized experience replay (PER). The proposed method demonstrates the

capability to predict non-healthy states almost one day before a fault occurrence, specifically between 17:34 and 19:20 on October 30th (while the performance and effectiveness evaluation was performed). The reliability of its performance is affirmed by available machine indices (MIs) and vibration indices (VIs). The DDQN-based FD method consistently classifies states before motor or reducer replacements as warning or alarm conditions. Comparatively, two other methods, Hidden Markov Model (HMM) and One-Class Support Vector Machine (OC-SVM), exhibit relatively good performance tailored to the case study but display stability issues. They often output different clusters for near-consecutive states, impacting the reliability of state classification. In contrast, the presented reinforcement learning (RL)-based method, utilizing DDQN, shows more stable trends and behavior closely aligned with the natural health status evolution of a machine. Furthermore, the DDQN-based FD method does not rely on ad-hoc pre-processing techniques or user-defined thresholds for output labels, making it adaptable to other FD domains and amenable to other RL algorithms. This flexibility enhances its potential applicability across various fault detection scenarios.

- Reinforcement Learning based on Stochastic Dynamic Programming for Condition-based Maintenance of Deteriorating Production Processes [20]

This paper presents the development of a stochastic dynamic programming model for maintenance planning in a deteriorating multistate production system. The quality of each bath/lot of items produced in each stage serves as a condition monitoring parameter for condition-based maintenance. The machine operates with m-1 operational states and a non-operational state, referred to as the failure state. At the beginning of each stage, four management actions are available: (1) renew the system; (2) implement maintenance; (3) continue production; and (4) inspect the system. The maintenance impact is considered imperfect, implying that after maintenance, the system is restored to any non-worse states with known probabilities. Since the system states change Markovianly at the end of each stage, and the quality of produced items depends on the system state, the system is modeled using a Markov decision process (MDP). Given that MDP is central to reinforcement learning, the paper discusses the application of the proposed stochastic dynamic programming for developing reinforcement learning, particularly for large-scale problems.

- Predictive Maintenance Decision Making Based on Reinforcement Learning in Multistage Production Systems [51]

While decision models for joint predictive maintenance and production in manufacturing systems are crucial, they remain largely unexplored. This paper proposes a novel decision model based on reinforcement learning, amalgamating production system modeling and approximate dynamic programming. The approach begins with developing a state-based model, analyzing the dynamics of a multistage production system with predictive maintenance. This model allows for a quantitative assessment of various disruptions and the impact of maintenance decisions on production. Subsequently, a reinforcement learning method is introduced to explore optimal maintenance policies that optimize both production and maintenance costs. To enhance the performance of the production system, machine stoppage bottlenecks are identified. An event-based indicator is employed for bottleneck identification using production data. Simulation case studies are conducted to test the proposed models, comparing them with three policies: state-based policy (SBP), time-based policy (TBP), and greedy policy (GP). The numerical studies reveal that the proposed decision

model outperforms these policies, demonstrating the lowest system cost. Specifically, it is 9.68%, 39.07%, and 39.56% lower than SBP, TBP, and GP, respectively. Additionally, the research underscores the significance of bottleneck identification and mitigation in achieving over a 9.00% throughput improvement in manufacturing systems.

- Joint optimization of preventive maintenance and production scheduling for multi-state production systems based on reinforcement learning [92]

In this study, the focus is on the integrated optimization of preventive maintenance and production scheduling for multi-state single-machine production systems with deterioration effects. The primary objective is to minimize the long-run expected average rewards, considering processing costs, maintenance costs, and completion rewards. To address this problem, the researchers establish a Markov Decision Process (MDP) model for the infinite-horizon expected average rewards, discussing the existence of an optimal stationary policy for the model. The R-learning algorithm is introduced as a solution to this long-run average expected reward problem. After analyzing the appropriate conditions for carrying out preventive maintenance under the optimal stationary policy, a novel HR-learning algorithm is presented, building upon the R-learning approach. Numerical results indicate that the proposed HR-learning algorithm outperforms R-learning and GR-learning methods. The performance analysis also explores the impact of the number of job types and states on the expected average rewards. Particularly in large solution spaces, HR-learning demonstrates a considerable impact of the number of job types on expected average rewards compared to R-learning and GR-learning. Computational results reveal that HR-learning algorithm stability surpasses that of R-learning for most cases. Additionally, the number of states has minimal impact on the algorithm's performance. This suggests a promising method for solving large-scale integrated optimization problems in practical production scenarios. However, the study notes the algorithm's sensitivity to state transition probabilities in certain cases. It is important to mention that the effects of different workpiece features on machine state deterioration are not addressed in this work.

- Deep Reinforcement Learning-based maintenance decision-making for a steel production line [81]

In this study, they propose a Deep Reinforcement Learning (DRL) policy designed for a steel production line that relies on scrap materials. The primary objective of the proposed policy is to make real-time maintenance decisions based on the monitored condition of the production line, with the aim of minimizing the long-term maintenance cost per unit of time. Specifically, the policy determines the optimal timing for performing preventive maintenance (PM) on the shredder machine, a critical operation affecting the entire steelmaking process. This decision is guided by factors such as the instantaneous productive rate of the shredder and the buffer level. To develop and fine-tune the PM policy, they constructed a simulation model that replicates the dynamic behavior of the production line. This model facilitates the training of the DRL agent through interactive sessions with the simulated environment. Subsequently, the performance of the PM policy is assessed through a comparative analysis, benchmarking it against other commonly used PM policies within the same context. The findings indicate that the DRL policy consistently outperforms alternative strategies, leading to a substantial reduction in maintenance costs (up to 67.5%). Moreover, the DRL policy effectively mitigates unmet demand and critical maintenance (CM) scenarios. This superior

performance is attributed to the DRL policy's capability to make maintenance decisions in real-time, responding promptly to the dynamic conditions rather than relying on scheduled PM actions. The policy's ability to suggest optimal times for PM actions is a direct result of understanding the dynamic nature of the environment. In conclusion, the outcomes of this research endorse the effectiveness of the DRL tool for making maintenance decisions. The DRL approach demonstrates a reduction in the expected long-term cost rate, coupled with enhancements in system availability and reliability. The integration of Artificial Intelligence (AI) and Machine Learning (ML) tools into maintenance, production control, and management emerges as a potent strategy for bolstering industry competitiveness.

	Time-based policy	Inspection-based policy	Corrective Policy	Proposed DRL Policy
Decision variables (hours)	$T = 49.53$	$T = 34.61$	-	-
Cost per unit of time (un. /hours)	90.996	33.973	106.7	29.57
Unmet demand per unit of time (ton. /hours)	0.6314	0.016	0.554	0.0

Figure11: Policies comparison [81]

- Demonstrating Reinforcement Learning for Maintenance Scheduling in a Production Environment [27]

This paper demonstrates the applicability of reinforcement learning (RL), specifically Q-learning, for devising an optimal strategy to schedule maintenance capacity within a practical production environment. The RL algorithm proves capable of learning diverse maintenance strategies contingent on distinct optimization objectives and predefined boundary conditions influencing machine degradation modes and maintenance costs. To facilitate the training of the RL algorithm under various conditions, the paper introduces a software-based plant model designed for discrete event simulation in a representative production setting. The proposed plant model emulates interconnected machines in a multi-stage, multi-product production process characteristic of modern manufacturing. These machines undergo degradation modeled as a Markov chain over time, necessitating maintenance for system output optimization. The plant model's building blocks, functionality, and interdependencies are crafted to align with conditions and restrictions prevalent in contemporary production environments. The object-oriented architecture of the plant model allows integration with real-time data from physical production environments, offering flexibility and scalability when adapting to new production configurations. Both the proposed plant model and the RL algorithms used in the simulation are designed to be adaptable to specific situations, environments, and maintenance measures. The overarching goal of this system is to enhance maintenance planning, particularly in unforeseen circumstances such as unplanned startup failures or production interruptions.

- Deep reinforcement learning based preventive maintenance policy for serial production lines [24]

This study addresses the challenging decision of when and where to perform preventive maintenance in a serial production line with intermediate buffers. The complexity and stochastic nature of such production lines make this decision nontrivial. To enhance the cost efficiency of serial production lines, the paper proposes a deep reinforcement learning-based approach to derive a preventive maintenance (PM) policy. The learning process involves a novel modelling method for the serial production line, and a reward function is introduced based on the evaluation of system production loss. The Double Deep Q-Network algorithm is applied for learning the PM policy. Simulation results demonstrate the effectiveness of the learning algorithm, showing an increased throughput and reduced cost. On average, the learned policy reduces the overall maintenance cost rate by 8.77% and 6.25% comparing to the age dependent policy and opportunistic policy respectively. Notably, the learned policy often involves 'group maintenance' and 'opportunistic maintenance,' concepts and rules that were not explicitly provided during the learning process, highlighting the effectiveness of the problem formulation, algorithm, and reward function proposed in the paper.

- Reinforcement learning for combined production-maintenance and quality control of a manufacturing system with deterioration failures [62]

This research paper focused on a stochastic system facing frequent degrading failures and aimed to determine optimal joint control policies that maximize the total profit of the system. The study implemented a decision-making framework where production, maintenance, and recycle control policies were derived through a reinforcement learning algorithm. Simulation experiments were conducted to evaluate the effectiveness of the approach. Results indicated that the agent effectively managed inventory levels by authorizing recycle and production activities. Maintenance activities were frequently authorized to prevent further degradation of product quality and maintain the functionality of the manufacturing facility. Additionally, alternative control policies were described and compared to the proposed approach, demonstrating that the integrated joint policy was more profitable than ad-hoc policies.

- Reinforcement Learning-Based and Parametric Production-Maintenance Control Policies for a Deteriorating Manufacturing System [3]

In this paper, a model for a stochastic production/inventory system subject to deterioration failures is developed and analysed. The system operates in an environment where customer interarrival times are random, allowing for backorders. The system goes through multiple deterioration stages before ultimately failing, with repair and maintenance activities restoring it to previous states. Both repair and maintenance durations are considered stochastic. The objective is to minimize the expected sum of two conflicting functions: the average inventory level and the average number of backorders. The challenge is to find an optimal trade-off between maintaining a high service level and minimizing inventory. To address this problem, the paper introduces a novel reinforcement learning-based approach for obtaining optimal or near-optimal joint production/maintenance control policies. The study also explores parametric production and maintenance policies commonly used in practical situations, including Kanban, (s, S), threshold-type condition-based maintenance, and periodic maintenance. Through extensive simulation experiments, the proposed reinforcement learning-based approach consistently outperforms parametric policies (Kanban – CBM, Kanban – PM, ((s, S) – CBM

and (s, S) - PM). The results of the experiments shed light on the behaviour of parametric policies and highlight the superior performance and structural insights derived from the reinforcement learning-based approach..

7.5.1. Pioneering Contributions to the State of the Art:

Reinforcement Learning (RL) has significantly reshaped the landscape of manufacturing maintenance strategies and quality control, as evidenced by a collection of ground-breaking articles. These contributions span a spectrum of applications, from optimizing maintenance decision-making in single-machine systems to addressing the complexities of joint maintenance and production scheduling in large-scale manufacturing networks. The versatility of RL is underscored as it navigates challenges in additive manufacturing defect mitigation, handles uncertainties in deteriorating production processes, and integrates seamlessly with preventive maintenance policies for multistage production systems. The articles also showcase RL's adaptability in real-world production environments, emphasizing its role in practical maintenance scheduling. Whether applied to steel production lines, serial production setups, or parametric manufacturing contexts, RL emerges as a promising tool for enhancing the efficiency and adaptability of maintenance and quality control strategies across diverse manufacturing domains. These pioneering contributions collectively propel RL to the forefront of research, guiding future endeavours toward more efficient and adaptive solutions tailored to the evolving challenges in manufacturing processes.

7.5.2. Advantages of Reinforcement Learning for Maintenance Strategies and Quality:

Reinforcement Learning (RL) offers distinct advantages in the context of maintenance strategies and quality control within the manufacturing sector. Its primary strength lies in the ability to learn optimal decision-making policies without the need for an explicit system model. This is particularly beneficial in manufacturing, where systems are often intricate and challenging to accurately model. RL excels in managing stochastic and dynamic environments, characteristics commonly encountered in manufacturing settings. In the realm of maintenance, RL has been successfully applied to diverse problems, including condition-based maintenance, preventive maintenance, and the integration of maintenance with production scheduling. Noteworthy examples, ‘A Reinforcement Learning Algorithm for Optimal Dynamic Policies of Joint Condition-based Maintenance and Condition-based Production [19]’, include the development of a reinforcement learning algorithm for optimizing dynamic policies in joint condition-based maintenance and production scheduling. Another application involved a deep reinforcement learning-based approach for decision-making in maintenance within a steel production line, ‘Deep Reinforcement Learning-based maintenance decision-making for a steel production line [81]’. These applications demonstrated that RL contributes to enhanced maintenance strategies and cost reduction. Similarly, RL has proven effective in addressing quality control challenges in manufacturing. For instance, a reinforcement learning-based approach was proposed for defect mitigation in additive manufacturing quality assurance, ‘Reinforcement learning-based defect mitigation for quality assurance of additive manufacturing [11]’. Another study introduced a deep multi-agent reinforcement learning approach for multi-level preventive maintenance in manufacturing systems, ‘Deep multi-agent reinforcement learning for multi-level preventive

maintenance in manufacturing systems [75]'. These initiatives showcased the capability of RL to enhance quality control and mitigate defects. In summary, RL stands out with its advantages over traditional methods in shaping maintenance strategies and ensuring quality control in manufacturing. Its capacity to learn optimal decision-making policies without explicit system modeling, along with its adaptability to stochastic and dynamic environments, positions RL as a valuable tool. The successful application of RL in various maintenance and quality control scenarios within manufacturing underscores its potential for improving performance and reducing costs.

7.5.3. Areas for improvement and future directions:

- further attention is needed to enhance the adaptability of the algorithm to diverse cyber-physical manufacturing environments.
- the need for the algorithm to handle real-time quality inspection data efficiently.
- refining the algorithm's adaptability to varying workloads and dynamic system states.
- further work is needed to handle complex dependencies between maintenance actions, production schedules and quality processes.
- addressing issues related to real-time decision-making in distributed settings.
- enhancing the algorithm's resilience to uncertainties and its ability to adapt to varying deterioration patterns.
- improving the algorithm's ability to dynamically adjust preventive maintenance schedules based on evolving production demands.
- improvements in the algorithm's robustness to harsh production conditions (especially in steel production).
- enhancements in the algorithm's adaptability to different production line setups
- Improvements in handling diverse deterioration patterns

The articles shed light on noteworthy gaps that warrant attention in future research. Firstly, an enhanced post-prognostics production planning and control (PPC) approach could be achieved by incorporating additional steps in the PPC process, such as raw materials procurement, scheduling, and dispatching. Secondly, addressing unknown demands and enhancing algorithm performance could be accomplished by integrating sales forecasts into the models. Extending models to encompass multi-component or multi-machine systems offers potential benefits, allowing for the adjustment of production levels to facilitate opportunistic maintenance by grouping interventions. There is considerable potential in establishing connections between prognostics and Reinforcement Learning (RL) algorithms. Prognostics algorithms could gain insight from RL agents about upcoming actions, considering the load exerted on the machine. Conversely, RL agents could benefit from prognostics by evaluating alternative actions, such as predicting machine breakdowns at different production levels. Exploring whether algorithmic complexity can be reduced is an avenue for investigation, including possibilities like directly feeding sensor data to RL agents without a prognostics model or employing simpler simulation-based optimization techniques like response surface or gradient descent methods. Another area for improvement lies in the hierarchical decision framework, which could be enriched with additional criteria and indicators, especially those related to energy consumption. Given the

growing emphasis on sustainability and energy efficiency, incorporating such considerations into decision-making frameworks is crucial. Future work with digital twin applications suggests the development of a more detailed digital twin model to serve as a realistic training environment for AI-based research on maintenance and quality inspection. Furthermore, in the realm of joint optimization of Manufacturing Systems (MS), future research should explore a more comprehensive approach. This could involve integrating aspects beyond maintenance, such as production scheduling, human reliabilities, and the maintenance of soft systems. A holistic joint optimization strategy could address factors like operator skill training and soft system maintenance, contributing to the reduction of MS failures caused by human errors or soft bugs.

7.6. Applications of Reinforcement Learning in Real-Time Demand Response for Sustainable Manufacturing

- Explainable multi-agent deep reinforcement learning for real-time demand response towards sustainable manufacturing [94]

This study introduces an explainable multiagent deep reinforcement learning (RL) method, named Decomposed Multi-Agent Deep Q-Network (DMADQN). The approach utilizes an analytical manufacturing system model to decompose the system-level energy management objective and production requirement to the agent level. By decomposing the task, the agent can create a safe action subset that is interpretable, aiming to fulfill the original system-level production requirement while learning to reduce energy costs under demand response (DR). The method is applied to control a section of an automotive assembly line using one year of DR electricity price data to validate its performance. Results indicate that the proposed DMADQN method ensures the achievement of the production requirement while delivering better DR energy management performance in both RL training and testing phases. Moreover, the proposed approach outperforms the day-ahead scheduling approach and achieves up to an additional 30.7% savings in energy costs under dynamic DR conditions.

- Demand Response Optimization of Cement Manufacturing Industry Based on Reinforcement Learning Algorithm [89]

In the context of industrial manufacturing striving to achieve carbon neutrality goals, optimizing energy efficiency is crucial. The study focuses on the energy-intensive cement manufacturing industry. It begins with a detailed modeling analysis of the main energy-consuming equipment in cement manufacturing based on industrial load characteristics. Subsequently, demand response scheduling methods for industrial settings, employing a reinforcement learning algorithm, are developed. Proximal Policy Optimization (PPO) is chosen as the reinforcement learning algorithm to implement industrial demand response. PPO is selected for its ability to mitigate the impact of large differences between old and new strategies during the training process on the learning process. Simulation experiments are conducted to verify the effectiveness and feasibility of the proposed scheme. The results indicate that the RL demand response scheduling scheme reduces the daily electricity cost for power users compared to scenarios without it. In the experiments, the daily cost with RL was 11,961.44, whereas without RL, the daily cost was 13,409.31.

- A reinforcement learning/ad-hoc planning and scheduling mechanism for flexible and sustainable manufacturing systems [63]

This paper introduces a reinforcement learning-based framework for optimizing the behavior of failure-prone machines integrated into a multi-stage production line processing a single type of product. The framework employs two agents at specific stages of the production process to plan various activities, including production and remanufacturing. Additionally, ad-hoc control policies related to production and maintenance are integrated to complement the reinforcement learning-based decision-making process. The objective is to enhance waste management, minimize redundant activity authorizations, and improve the overall system performance. Simulation experiments were conducted to assess the functionality of the proposed approach, and the results indicated that the manufacturing/remanufacturing system's revenue stream was primarily derived from recycled and remanufactured products, emphasizing the success of the green manufacturing strategy implemented through the reinforcement learning/ad-hoc control mechanism.

7.6.1. Pioneering Contributions on the State of the Art:

Let's explore the pioneering contributions related to the use of RL in sustainable manufacturing, considering the specified articles. "Explainable Multi-Agent Deep Reinforcement Learning for Real-Time Demand Response Towards Sustainable Manufacturing [94]" focuses on multi-agent deep reinforcement learning for real-time demand response, aiming to optimize energy management and provide explainable policy-level contrastive explanations for multi-agent RL. This work contributes to sustainable practices by addressing the complex dynamics of manufacturing systems. In "Demand Response Optimization of Cement Manufacturing Industry Based on Reinforcement Learning Algorithm [89]" the study investigates energy efficiency optimization in cement manufacturing using reinforcement learning. The approach minimizes electricity costs while maintaining production tasks, showcasing the feasibility of achieving cost-effective, green, and sustainable manufacturing in cement plants. "A Reinforcement Learning/Ad-Hoc Planning and Scheduling Mechanism for Flexible and Sustainable Manufacturing Systems [63]" addresses process scheduling in manufacturing, emphasizing long-term sustainability by combining reinforcement learning with ad-hoc manufacturing/maintenance control. Simulation experiments validate improved process planning, inventory management, and cost-effective sustainable practices. Together, these articles significantly contribute to advancing the field of sustainable manufacturing, leveraging RL techniques to enhance energy efficiency, reduce costs, and promote environmentally conscious practices.

7.6.2. Advantages of RL in Sustainable Manufacturing:

Advantages of RL in the context of sustainable manufacturing, drawing insights from the specified papers are the following: benefits in energy efficiency by enabling real-time demand response and dynamically adjusting energy usage, contributing to sustainability. Multi-agent RL enhances scalability by allowing coordination among various manufacturing entities and efficiently adapting to changing demand patterns. One of the papers emphasizes the importance of explain ability through policy-level contrastive explanations for Multi-Agent Reinforcement Learning (MARL), aiding practitioners in understanding and fine-tuning RL policies. Another work showcases RL's capacity to optimize demand

response scheduling, reducing electricity costs while maintaining production tasks. RL ensures efficient resource utilization, mitigating waste and environmental impact, and adapts to dynamic conditions, promoting long-term sustainability by responding to market fluctuations and production variations. The last paper highlights RL's role in enhancing manufacturing system flexibility, dynamically adjusting schedules to accommodate changes in demand, maintenance, and resource availability. Integrating RL with ad-hoc control considers economic, environmental, and social aspects, contributing to the triple-bottom-line sustainability—balancing profit, planet, and people. The paper emphasizes the need to assess the impact of disposition options on sustainability, providing a holistic view through RL that considers diverse factors. In summary, RL empowers sustainable manufacturing by optimizing energy consumption, reducing costs, and promoting environmentally conscious practices.

7.6.3. Areas for improvement and future directions:

Considering the paper that focuses on the application of RL in sustainable manufacturing, the principal areas for improvement are:

• Real-time Data Handling: The need for RL algorithms to handle real-time data efficiently, particularly when responding to dynamic demand scenarios in sustainable manufacturing.

• Dynamic Adaptability: Refining RL algorithms' adaptability to varying demand patterns and dynamic manufacturing environments for more effective real-time responses.

• Scalability: Further work is needed to ensure the scalability of multi-agent deep RL algorithms for large-scale sustainable manufacturing systems.

• Handling Complex Dependencies: Enhancing the ability of RL algorithms to handle complex dependencies between real-time demand response actions, production schedules, and control processes.

The challenges and future directions of RL in the context of sustainable manufacturing, drawing insights from the specified papers, are the following. In "Explainable Multi-Agent Deep Reinforcement Learning for Real-Time Demand Response Towards Sustainable Manufacturing [94]" challenges include the black-box nature of deep RL models, hindering interpretability, coordinating multiple agents in manufacturing environments, and ensuring real-time adaptation to changing conditions. Future directions suggest hybrid approaches by combining deep RL with other techniques, integrating human expertise into RL systems, and leveraging transfer learning for specific manufacturing tasks. "Demand Response Optimization of Cement Manufacturing Industry Based on Reinforcement Learning Algorithm [89]d" faces challenges in handling the intricate processes and nonlinear dynamics of cement manufacturing, balancing energy efficiency and production targets, and generalizing RL models to diverse plant conditions. Future directions propose hierarchical RL architectures, domain-specific exploration strategies, and incorporating safety constraints into RL policies. In "A Reinforcement Learning/Ad-Hoc Planning and Scheduling Mechanism for Flexible and Sustainable Manufacturing Systems [63]" challenges involve adapting to dynamic manufacturing environments,

efficiently allocating resources while considering sustainability goals, and addressing the computational complexity of scaling RL approaches to large systems. Future directions advocate for multi-objective RL to balance conflicting objectives, decentralized RL for resource allocation and scheduling, and designing robust RL policies that perform well under uncertainty and disturbances. In summary, the future of RL in sustainable manufacturing lies in addressing challenges related to interpretability, coordination, real-time adaptation, and domain-specific nuances, while exploring hybrid approaches, transfer learning, and human-in-the-loop RL.

8. RL algorithms' classification:

Considering all the algorithms that are used in the papers and that are listed in Table 1 it is important to better understand them by classifying them. In RL the agents are trained according to the generalized policy iteration principle. Herein two stages are distinguished, namely policy evaluation and policy improvement. During policy evaluation, agents select actions according to a policy function, and observe their returned reward. During policy improvement, the policy is adjusted based on the observations made. These stages are repeated until convergence (hopefully). RL algorithms can be classified along three discrete axes: value, policy or actor-critic methods (1), on- or off-policy methods (2), and model-free or model-based methods (3).

(1) **Value, Policy and Actor-Critic Methods:** Value methods try to estimate future reward by means of a value function, which is used to estimate the "goodness" of either states or actions from given states. During the policy evaluation stage, actions are selected by using the value function to ascertain the quality of the state's reachable from the current one. The observed rewards are used during the subsequent stage to improve the value function. Examples of such methods are State Action Reward State Action (SARSA), Q-Learning (QL), Deep Q-learning (DQN), and Double DQN (DDQN). QL is particularly popular with the production scheduling community with implementations using tables to represent the value functions that are used, or neural networks as function approximators. Basic off-policy value-based approaches like Q-Learning show better sample efficiency but are often unstable when integrated with function approximators. Alternatively, agent policies can be used directly to select actions during the evaluation stage. Based on the observed rewards, the reward expectation under policy is estimated, its gradient with respect to is computed and the policy is updated using stochastic gradient ascent. Examples include RE-INFORCE used for production scheduling, they can naturally handle continuous state and action spaces and learn stochastic policies and are sample inefficient and show poor robustness. In recent years, algorithmic advances have been made to address the above-mentioned deficits. The development of the so-called TRPO and PPO algorithms led to further improvement of sample efficiency and robustness when using policy-based approaches [35].
To combine the advantages of policy-based and value-based method while minimizing their shortcomings, current research efforts include actor-critic methods [35]. Instead of using environment interaction to approximate the expected reward directly, the (state) value function approximator (critic), is used to inform the policy approximator (actor) of the quality

of its action. AlphaZero (AZ) and Deep Deterministic Policy Gradient (DDPG) fall in this category.

(2) One of **On- vs Off-Policy:** On policy methods (e.g. SARSA) use the same policy during evaluation stage that was adjusted during the improvement stage. This leads to more stable learning at the expense of exploration, which can lead to local optima. Conversely, in off-policy methods (e.g. QL, DQN, DDQN), the policy used during the evaluation stage can differ from the one used in the improvement stage, which leads to more exploration at the expense of convergence speed.

(3) **Model Based vs. Model Free:** RL algorithms can be furthermore split into model-free and model-based approaches. Model-based approaches use an environment model to plan a few steps into the future before deciding on an action. The involved environment model is either estimated by the agent itself, e.g. Imagination Augmented Agent (I2A), or simply given to it, e.g. AZ. Single- vs Multi-Agents: multiple agents are allowed to act within the same environment. These agents can be cooperative, i.e. striving to jointly maximize the expected reward or competitive, with each agent targeting a maximization of his reward only. For production scheduling multi-agent systems are often deployed. Depending on the MDP breakdown, agents can be associated with different setup components [71]. In constraint, model-free algorithms directly learn a policy or value function from the environment without explicitly modelling the dynamics of the environment. These algorithms learn by interacting with the environment, receiving feedback in the form of rewards, and adjusting their policies or value estimates accordingly. DQL and DDQN fall into this category, as they both belong to the broader family of Q-learning algorithms, which are known for being model-free.

Algorithm DRL-based scheduling of decentralized robot services with DDQN algorithm

Initialize learning rate l, minibatch size b, discounted factor γ, maximal exploration value ε
Initialize replay memory D with capacity N
Initialize Q-network θ, α, β and target-network parameters $\theta^{-}, \alpha^{-}, \beta^{-}$
for each episode do
  for each step t (task O_k) do
  Reset cloud manufacturing environment to initial state $s = s_t$
  With probability ε select a random action (service) $a = a_t$
  Otherwise select $a_t = argmax_a Q(s_t, a_t; \theta, \alpha, \beta)$
  Schedule action a_t, observe reward r_t and next state (next order O_{k+1}) $s' = s_{t+1}$, $s_t = s_{t+1}$
  Store transition $(s_t,\ a_t,\ r_t, s_{t+1})$ in D
  if episode terminates at step $j+1$ then
  set $y_j = r_j$
  else
  $y_j = r_j + \gamma max_{a'} Q(s_{j+1}, a'; \theta^{-}, \alpha^{-}, \beta^{-})$
  end if
  Update Q-network parameters $\theta, \alpha,\ and\ \beta$ with a loss function of $L(\theta, \alpha, \beta) = \frac{1}{N}[(y_j - Q(s, a; \theta, \alpha, \beta))^2]$
  Compute TD-error $\delta_j = y_j - Q(s_t, a_t; \theta, \alpha, \beta)$
  Every C steps reset $\theta^{-}, \alpha^{-}, \beta^{-} \leftarrow \theta, \alpha, \beta$
  end for
end for

Figure 12: Pseudo-code of the DDQN-based scheduling algorithm of decentralized robot services.

33 out of 98 works used DQL or DDQL which are value, off-policy and model free algorithms. They are applied in all the areas even though they are mostly used in scheduling problems (25 papers in Scheduling, 2 in Process control, 4 in Maintenance and quality, 1 in Sustainable Manufacturing and 1 in Autonomous Manufacturing). "Scheduling of decentralized robot services in cloud manufacturing with deep reinforcement learning [49]" is an example of this algorithm in scheduling problems. The pseudo code is shown in Figure 12.

Different from DQN, DDQN modifies the network architecture, and the Q value is divided into two parts: state value and action advantage, which can be described by:

$$Q\pi(s, a) = V\pi(s) + A\pi(s, a)$$

DDQN not only evaluates the value $Q\pi(s, a)$ of an action in a certain stable convergence results, the average method replaces the maximum method and can be defined by:

$$Q\pi(s, a, \vartheta, \alpha, \beta) = V\pi(s; \vartheta, \alpha) + (A\pi(s, a; \vartheta, \beta) - (1/|A|)\Sigma a' A\pi(s, a' ; \vartheta, \beta))$$

where $|A|$ is the number of a discrete action set. The advantage function is close to the mean and training can obtain stable convergence results [49].

General definitions:

- Learning Rate (l): This is a hyperparameter in the RL algorithm that determines the size of the steps taken during the weight updates of the Q-network. It is a crucial parameter in determining how quickly or slowly the Q-network adapts its weights to the training data, impacting the stability and speed of convergence during the training process.
- Q-network: In RL, a Q-network is a neural network that learns to approximate the Q-values, which represent the expected cumulative future rewards for taking a particular action in a given state.
- Weight Update: During the training of a neural network, including the Q-network, the weights of the network are adjusted to reduce the difference between the predicted output and the actual target (in the case of RL, the Q-values). This adjustment is typically done using an optimization algorithm like gradient descent.
- Gradient Descent: This is an optimization algorithm used to minimize the error in the neural network. The learning rate controls the size of the steps taken during the descent. A larger learning rate means larger steps, but it can risk overshooting the minimum; a smaller learning rate takes smaller steps but may take longer to converge.

Now let's break down the steps of the DDQN algorithm shown in Figure 12.

1. Initialization:
    - Learning Rate (l): This parameter controls how much the Q-network's weights are updated after each iteration. It determines the step size during gradient descent.
    - Minibatch Size (b): The number of transitions sampled from the replay memory to update the Q-network at each step.
    - Discounted Factor (γ): A value between 0 and 1 that discounts future rewards. It balances immediate rewards versus long-term rewards.
    - Maximal Exploration Value (ε): The probability of selecting a random action (exploration) instead of the best action (exploitation).
2. Replay Memory Initialization (D):
    - A buffer that stores past experiences (transitions) for training the Q-network.
3. Q-Network Initialization:
    - θ, α, β: Parameters of the Q-network.
    - θ^-, α^-, β^-: Parameters of the target network (used for stability during training).
4. Episode Loop:
    - For each episode, the following steps are executed:
5. Step Loop:
    - For each step within an episode (task Ok):
        - Reset the cloud manufacturing environment to its initial state (s = st).
        - With probability ε, select a random action (service) a = at. Otherwise, select at as the action that maximizes the Q-value based on the current Q-network parameters (θ, α, β).
        - Schedule the selected action at, observe the reward rt, and transition to the next state (next order Ok+1) s' = st+1 (updating st).
        - Store the transition (st, at, rt, st+1) in the replay memory D.
6. Termination Check:
    - If the episode terminates at step j + 1:
        - Set yj = rj (the immediate reward).
    - Otherwise:
        - Compute yj = rj + γ * maxa' Q(sj+1, a'; θ , α, β) (the discounted future reward).
7. Q-Network Update:
    - Update the Q-network parameters (θ, α, β) using the loss function:
        - L(θ, α, β) = 1/N * Σ[(yj - Q(s, a; θ, α, β))^2].
    - Compute the TD-error δj = yj - Q(st, at; θ, α, β).
8. Target Network Update:
    - Every C steps, reset the target network parameters (θ^-, α^-, β^-) to the current Q-network parameters (θ, α, β).

Let's break down the Q-learning algorithm in simpler terms:

1. Environment and Agent:
   - Imagine a robot (the agent) exploring a new world (the environment).
   - The robot wants to learn how to take actions (like moving left, right, or picking up objects) to maximize its rewards (like finding the right job or avoiding danger/collusions).
2. Q-Values:

- The robot keeps track of a special value for each state-action pair. We call this value the Q-value.
- The Q-value represents how good it is to take a specific action in a particular state.
- Initially, the robot doesn't know anything, so all Q-values are random.

3. Exploration vs. Exploitation:
   - The robot faces a dilemma:
     - Exploration: It can try new actions to learn more about the environment.
     - Exploitation: It can choose the action with the highest known Q-value.
   - Balancing exploration and exploitation is crucial.
4. Learning Process:
   - The robot explores by taking actions randomly or based on some exploration strategy (like flipping a coin).
   - It observes the reward it gets for each action and the new state it ends up in.
   - It updates its Q-values using a formula that combines the observed reward and the Q-value of the next state.
5. Updating Q-Values:
   - The robot adjusts its Q-values based on the observed rewards:
     - If the reward was good, it increases the Q-value for that action.
     - If the reward was bad, it decreases the Q-value.
   - The robot also considers the best Q-value of the next state (using a discount factor) to make its decision.
6. Repeat and Improve:
   - The robot keeps exploring, taking actions, and updating Q-values.
   - Over time, it learns which actions lead to better rewards.
   - Eventually, it becomes smarter and chooses actions that maximize its total reward.
7. Target Network:
   - To stabilize learning, the robot maintains a separate "target" Q-network.
   - Every so often, it updates the target network with the current Q-values.
8. Goal:
   - The robot's goal is to find the best actions for each state so that it can navigate the environment effectively and collect maximum rewards.

This is a simplified explanation, but it captures the essence of how Q-learning works. The robot learns from experience, adjusts its Q-values, and becomes better at making decisions over time.

This algorithm aims to learn an optimal Q-function that estimates the expected cumulative reward for taking a specific action in a given state. It balances exploration (trying new actions) and exploitation (choosing the best-known action) to improve decision-making in the cloud manufacturing environment. The Q-network is updated iteratively based on observed transitions and rewards. This algorithm involves initializing parameters and networks, interacting with the environment, storing experiences in the replay memory, updating the Q-network based on sampled transitions, and periodically updating the target network to stabilize training. The loss function is based on the temporal difference (TD) error, which measures the discrepancy between the predicted Q-value and the target Q-value. The overall goal is to train the Q-network to accurately estimate Q-values and improve decision-making in the cloud manufacturing environment.

8.1. Use case algorithm development analysis

Among the many selected papers, I've decided to describe the paper number 70 in Table 1, "Designing an adaptive production control system using reinforcement learning" as it is possible to access to the open-source repository SimRLFab (Kuhnle 2020). Infact, this repository contains the simulation as well as RL-agent framework for order dispatching in a complex job shop manufacturing system that is described in the work.

Introduction:
Highly dynamic and complex production systems, manufacturing characteristics of the semiconductor wafer fabrication, challenge manufacturers on optimal production control solutions that can satisfy rising customer requirements. The paper addresses the design of RL to create an adaptive production control system by the real-world example of order dispatching in a complex job shop.
A job shop consists of several machines (processing resources) that process jobs (products, orders) based on a defined list or process steps. After every process, the job is dispatched and transported to the next processing machine. Machines are usually grouped in sub-areas by the type processing type, i.e. similar processing capabilities are next to each other.
In operations management, two tasks are considered to improve operational efficiency, i.e. increase capacity utilization, raise system throughput, and reduce order cycle times. First, job shop scheduling is an optimization problem which assigns a list of jobs to machines at times. It is considered as NP-hard due to the large number of constraints and even feasible solutions can be hard to compute in reasonable time. Second, order dispatching optimizes the order flow and dynamically determines the next processing resource. Depending on the degree of stochastic processes either scheduling or dispatching is enforced. In manufacturing environments with a high degree of unforeseen and stochastic processes, efficient dispatching approaches are required to operate the manufacturing system on a robust and high performance [6].
This framework provides an integrated simulation and reinforcement learning model to investigate the potential of data-driven reinforcement learning in production planning and control of complex job shop systems. The simulation model allows parametrization of a broad range of job shop-like manufacturing systems. Furthermore, performance statistics and logging of performance indicators are provided. Reinforcement learning is implemented to control the order dispatching and several dispatching heuristics provide benchmarks that are used in practice [6].
The objective of the work is the development of an adaptive order dispatching optimizing material handling routes under consideration of machine utilization and order throughput time that, at the same time, does not require a considerable amount of domain expertise, to allow the converge of the algorithm.
To exhibit real-world production characteristics, the discrete event simulation model is parameterized with historical data and assumptions on stochastic probability distributions.

Throughout the simulation, various actions are executed concerning the processing, release, and transportation of orders. These critical decision points are intricately tied to the dynamic nature of the production environment. Notably, changes in buffer levels occur when machines complete processing tasks or when the dispatcher concludes transportation activities. Each instance of such a change prompt all currently idle production resources to reevaluate and determine their subsequent actions based on the updated production state. This paper centrally focuses on elucidating these decision-making processes.

In the context of machine operations, the criterion for selecting the next order for processing adheres to a First-In-First-Out (FIFO) rule. In other words, the order that has been waiting the longest in the entry buffer is prioritized. Conversely, the determination of which order to dispatch next is entrusted to a Reinforcement Learning (RL) agent. Following the resolution of decisions by each resource regarding their next actions, the simulation seamlessly executes these choices, perpetuating the ongoing production workflows.

In establishing an adaptive production control system, a singular RL-agent assumes the pivotal role of determining the subsequent action for the dispatcher. Consequently, this agent emerges as the primary decision-maker for the dispatcher and is subsequently referred to as the RL dispatching agent. It's worth noting that the modeling approach remains independent of the quantity of dispatchers (instances) present in the system. This independence arises from the fact that the decision-making process remains fundamentally identical for any dispatcher, even in scenarios involving multiple dispatchers. Regardless of the count, the state information crucial to decision-making and the evaluation based on the reward signal remain consistent across all instances.

Authors chose the TRPO agent because of its robustness and to enhance the agent's ability to discern between valid and invalid actions, two key parameters are employed: the maximum recursion number and waiting time, configured at 5 and 2, respectively. These parameters play a critical role, particularly during the convergence of the agent. As the agent converges, the influence of these parameters on the production environment diminishes, given that the number of invalid actions becomes negligible in the converged state. Importantly, these last two parameters act as safeguards, preventing the agent from becoming ensnared in a loop of repeatedly choosing invalid actions. Their presence ensures that the agent avoids potential pitfalls and continues to make meaningful decisions in the dynamic production environment.

Parameter	Default configuration
Agent	TRPO[a]
Learning rate l	0.001
Discount rate γ	0.9
Network	f[b] x 128 x 128 x e[c]
Network activation	tangens hyperbolicus
Episode design	100 valid actions
Action mapping	direct mapping
Valid actions	$A_{S\rightarrow M}$, $A_{M\rightarrow S}$
Maximum recursion count	5
Waiting time t_w	2

Figure 13: Default configuration parameters of the used RL dispatching [35]

Performance indicators:

A simulation experiment entails numerous stochastic processes, where reproducibility relies on controlling each random number through a seed value. Despite these measures, inherent randomness persists due to the "black box" behavior of the RL-agent, which remains beyond control. Consequently, multiple simulation runs with identical configurations are executed, with preliminary studies indicating that three runs per configuration yield a suitably small confidence interval for quantitative comparisons. Throughout each simulation run, various performance indicators are recorded, and upon completion of the experiment, these recordings are scrutinized for comparison and evaluation. The key metrics encompass the reward assigned to the agent, average machine utilization (U) excluding downtimes, average waiting time of orders (WT), utilization of the dispatching agent, throughput of the entire production system, average inventory level (I), and the Alpha value (α). For the purposes of this paper, it suffices to understand Alpha as a measure influenced by the flow factor and machine utilization. By combining various performance indicators, Alpha evaluates the production system's performance, with a lower Alpha signifying better performance. However, a comprehensive understanding of achieved performance necessitates a detailed examination of key figures. To achieve this, raw values of the recordings undergo processing and summarization, including the computation of moving averages and standard deviations for each performance indicator across individual simulation runs and their combination. Additionally, the agent's convergence time is calculated, defined as the point when the moving average of the reward signal varies within a specified threshold range relative to the value of the reward's moving average. Final performance evaluation values are derived from the convergence point onward, excluding the training period, and serve as the primary basis for comparing different configurations.

Evaluation setup:

To enhance operational performance, the agent tackles two distinct challenges encapsulated within the term "order dispatching": order sequencing and route planning. The significance of these problems varies based on the production scenario, with the dispatching agent responsible not only for determining its next move but also for selecting the machine to handle the order from the available machines in the same group. The relative importance of these challenges depends on factors like transport resource limitations or machine bottlenecks in the system. When the transport resource acts as the bottleneck, optimized route planning is crucial. Conversely, if the transport resource's capacity is abundant, and machines pose the bottleneck, effective order sequencing takes precedence.

To investigate both scenarios, each RL-agent configuration undergoes testing in two distinct production scenarios, each emphasizing a specific problem. In the first scenario, the dispatching agent, acting as the system's transport resource, operates at a relatively slow speed (factor 0.3), and machine entry and exit buffers are limited (factor 0.5), spotlighting the importance of route planning. In contrast, the second scenario features a faster agent (factor 1.0) and larger buffers (factor 1.0), directing attention towards optimizing order sequencing for optimal machine utilization.

Performance indicators for rule-based benchmark heuristics are presented in Table 5 for both scenarios, evaluated based on average machine utilization (U), average order waiting time (WT) in an arbitrary time unit (TU), average inventory level (I), and the α-factor. Generally, Scenario 1 exhibits lower average inventory due to the slower agent and limited buffer capacity, resulting in lower machine utilization. However, the waiting time is reduced as orders are processed and transported more swiftly. Performance insights highlight that the RANDOM heuristic performs poorly, while VALID, FIFO, and NJF achieve better results. FIFO excels in minimizing average waiting time, particularly in Scenario 2 with extended waiting times, whereas NJF efficiently utilizes the bottleneck transport resource in Scenario 1, resulting in higher machine utilization.

The benchmark heuristics reveal a trade-off between machine utilization and waiting time. While NJF achieves higher machine utilization, it corresponds to an increase in average waiting time. On the contrary, applying a FIFO heuristic lower waiting time but reduces machine utilization. This trade-off emphasizes the rationale for using an RL-agent, aiming to leverage available information for multi-criteria optimization and concurrently minimize conflicts arising from inherent trade-offs.

In the process of learning an optimal order dispatching strategy, the RL-agent addresses order sequencing and route planning, engaging in a two-phased learning process. Initially, it learns to distinguish between valid and invalid actions, followed by learning the interplay between state information, selected actions, and rewards to optimize performance indicators.

Heuristic	Scenario 1				Scenario 2			
	U (%)	WT	I	α	U (%)	WT	I	α
RANDOM	41.8	182.2	8.5	27.6	38.7	180.4	**15.7**	16.2
VALID	56.5	86.5	**6.1**	3.74	83.9	115.6	20.2	0.69
FIFO	69.0	**81.8**	6.9	1.94	84.5	**115.0**	19.9	**0.56**
NJF	**81.7**	92.2	12.3	**0.73**	**84.7**	123.9	23.2	0.62

Bold values indicate the best, i.e. highest or lowest value in the column

Figure 14: Results for different rule-based heuristic dispatching approaches in both production scenarios

RL results' evaluation:

Authors decided to compute the algorithm performance by running different simulations to evaluate its capabilities and its relationship with different variables.

They started by evaluating if the agent was able to distinction between valid and invalid actions, then they computed the results for RL-agents with varying state information and reward signals, aiming to optimize specific production performance indicators – maximize the average machine utilization and lower the average waiting time of orders in the production system. The next evaluation was on results for RL-agents with fixed state information and reward functions when varying the episode design. After that they computed the results for RL-agents with fixed state information and reward signal when varying the action mapping as well as the set of actions the agent can execute. Next, they evaluated the RL-agents with fixed state information and the reward functions while varying weighting factors of the action subsets (shown in figure 15 as agent 31 and 32 outperformed the benchmarks in both scenarios). Lastly, they evaluated the results for RL-agents with fixed state information and a multi-criteria reward functions when varying weighting factors of the multi-criteria reward function.

Agent	State	Reward	Weighting factor	Scenario 1				Scenario 2			
				U (%)	WT	I	α	U (%)	WT	I	α
31	S^a	R_{w-uti}	$\omega_1 = 0.25, \omega_2 = 0.75$	77.1	**69.4**	**4.8**	0.93	84.8	**109.5**	**18.8**	0.57
32	S^a	R_{w-uti}	$\omega_1 = 0.5, \omega_2 = 0.5$	**82.0**	88.1	11.2	**0.73**	**86.9**	119.7	22.3	**0.47**
33	S^a	R_{w-uti}	$\omega_1 = 0.75, \omega_2 = 0.25$	78.1	120.7	19.7	1.54	86.1	123.7	24.4	0.51
34	S^b	R_{w-wt}	$\omega_1 = 0.25, \omega_2 = 0.75$	51.5	**71.8**	**4.2**	3.4	78.7	**106.7**	**18.7**	0.86
35	S^b	R_{w-wt}	$\omega_1 = 0.5, \omega_2 = 0.5$	53.7	80.0	7.7	**2.93**	78.1	108.1	20.2	0.76
36	S^b	R_{w-wt}	$\omega_1 = 0.75, \omega_2 = 0.25$	**58.3**	157.7	20.8	5.12	**86.6**	125.9	24.9	**0.56**

Bold values indicate the best, i.e. highest or lowest value in the column

[a] $S_{AS}, S_L, S_{MF}, S_{RPT}, S_{BEN}, S_{BEX}, S_{BPT}, S_{AT}$

[b] $S_{AS}, S_L, S_{MF}, S_{RPT}, S_{BEN}, S_{BEX}, S_{BPT}, S_{AT}, S_{WT}$

Figure 15: Results for RL-agents with fixed state information and the reward functions and while varying weighting factors of the action subsets [35].

Figure 16 presents a comprehensive summary of utilization and waiting time performance indicators for all RL-agents and benchmark heuristics through a two-dimensional scatter plot. This visualization effectively illustrates the potential of RL-agents in contrast to rule-based heuristics. Firstly, RL affords more detailed adjustments of desired performance, allowing for a broader spectrum of operation states in the production system. Secondly, the performance of heuristics exhibits significant variation when the scenario changes. Notably, the location and ranking of heuristics in the scatter plot alter with scenario shifts, emphasizing their lack of consideration for system bottlenecks. The highlighted RL-agents, specifically 31 and 32, demonstrate robustness across scenarios and consistently outperform other RL-agents and all heuristics in terms of both performance indicators, detailed results are shown in Table 15.

Expanding beyond individual scenarios, Figure 17 illustrates the moving average trends of machine utilization (U) and average waiting time (WT) for the heuristics FIFO and NJF, along with RL-agent 17. This agent, chosen for its simplicity and excellent performance in both scenarios, initially operates in a system parameterized according to Scenario 1 before transitioning to Scenario 2 after 10 million steps. As the scenario changes, both heuristics and the agent converge to performance values characteristic of Scenario 2, aligning with expectations for static heuristics that deterministically select actions. However, RL-agent 17 achieves nearly identical performance values compared to separate training in Scenarios 1 and 2, with only slightly better waiting times when scenarios change. This observation leads to the conclusion that RL-agents exhibit adaptability to changing production conditions without requiring a significant training phase. The performance indicators of the RL-agent adjust over the same number of steps as heuristics need for their performance adaptation. Additionally, training the agent in Scenario 1 and subsequently transitioning to Scenario 2 has a negligible effect on the final performance.

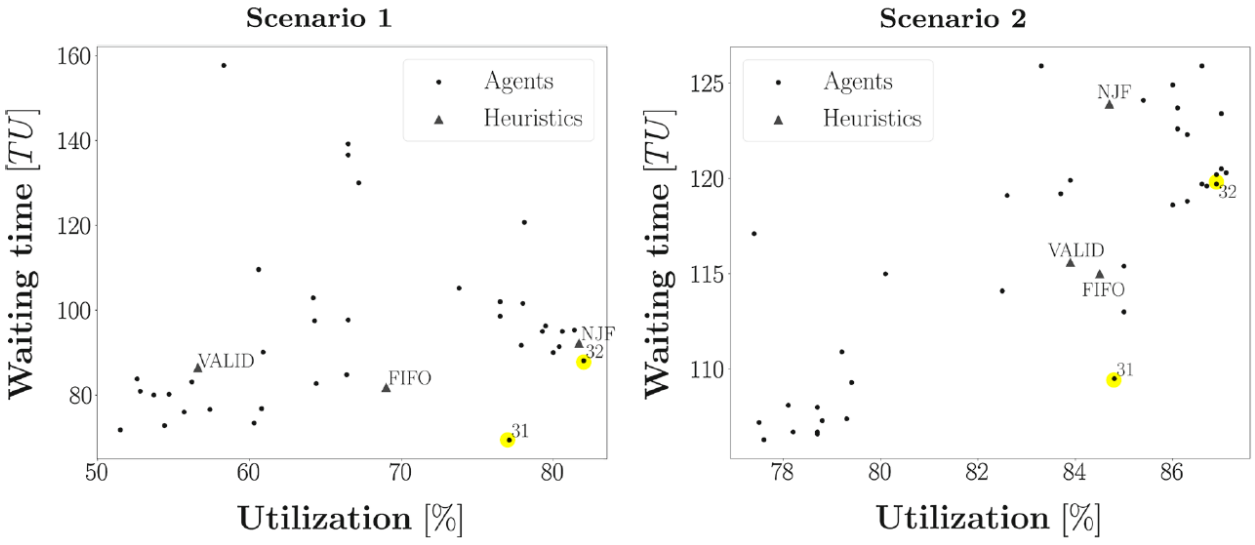

Figure 16: The scatter plot summarizes the waiting time and utilization performance of all heuristics and RL-agents presented in this paper. RL-Agents 31 and 32 are highlighted to show their superior performance in both scenarios [35].

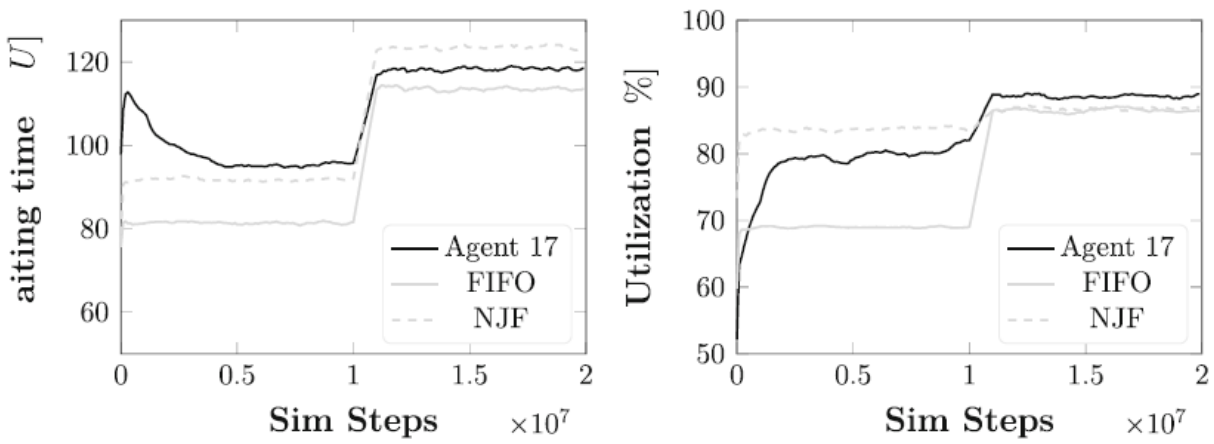

Figure 17: Machine utilization and average waiting time of orders when changing from Scenario 1 to Scenario 2 after 10 million simulation steps. Displayed are the heuristics NJF and FIFO as well as Agent 17 [35].

Comparison of time consumption when training RL-agents:

The experiments were executed on a Linux system featuring an Intel Xeon E5-2698 v4 CPU with 20 cores running at 2.2 GHz, 256 GB RDIMM DDR4 system memory, and an SSD for storage. The simulation environment and RL-agents were implemented in Python 3.6, utilizing the simply and tensorforce packages.For agents rewarded with a dense reward function, the computation of one million simulation steps takes approximately 1 hour. As the simulation duration increases, a slight decrease in computation speed is observed due to heightened data handling and recording efforts. Sparse agents, in contrast, demand roughly three times the computation time due to the increased complexity of neural network updates.In general, RL-agents converge faster in Scenario 2 compared to Scenario 1. Simple agents, require three to five million simulation steps in Scenario 1, whereas in Scenario 2, convergence is achieved after one to three million steps. Introducing additional state information and employing more complex reward functions results in increased time consumption. Complex agents, necessitate 20 to 25 million simulation steps in Scenario 1 and 10 to 13 million steps in Scenario 2 to reach convergence.An exception to these general patterns is agent 12, which required 45 million simulation steps to converge. The extended convergence time for this agent can be attributed to the combination of state and reward. Although the agent received state information on the current waiting time of orders, it initially did not correlate with the reward signal Ruti. However, the agent eventually discovered a correlation.

Algorithm description:

Reinforcement learning is applicable to optimization problems that can be modelled as sequential decision-making processes, i.e., Markov Decision Processes (MDP), therefore the problem must be modelled as an RL task by defining the agent, environment, actions, policy and rewards.

Let’s break down the variables modelled in the studied work:

The agent:

```
{
    "agent": "tensorforce",
    "update": 4,
    "objective": "policy_gradient",
    "reward_estimation": {
        "horizon": 2
    }
}
```

Figure 18: Modelled agent [6]

- "agent": "tensorforce": This specifies the choice of the reinforcement learning agent, and in this case, it's set to "tensorforce," indicating that the Tensorforce library will be used.
- "update": 4: This parameter likely refers to the number of update steps or iterations during the training process. The agent's parameters are adjusted based on the collected experience from the environment.
- "objective": "policy_gradient": This defines the objective, or the optimization method used during training. In this case, it's set to "policy_gradient," indicating that the training will involve optimizing the policy using gradient-based methods.
- "reward_estimation": { "horizon": 2 }: This specifies a parameter related to how rewards are estimated. The "horizon" parameter is set to 2, which might refer to a temporal horizon for estimating rewards. In reinforcement learning, the temporal horizon often represents the number of time steps into the future for which rewards are considered.

The environment:

```
{
    "environment": "gym",
    "level": "production-v0",
    "max_episode_timesteps": 100
}
```

Figure 19: Modelled environment [6]

- "environment": "gym": This line specifies the choice of the environment for the reinforcement learning task. In this case, it's set to "gym," indicating that the OpenAI Gym library will be used to set up the environment. OpenAI Gym provides a variety of environments to test and develop reinforcement learning algorithms.
- "level": "production-v0": This parameter likely specifies a particular environment within the OpenAI Gym toolkit. The environment is labeled as "production-v0." In OpenAI Gym, environments are typically labeled with a version number to indicate different configurations or variations of a specific task. The exact meaning of "production-v0" would depend on the specific environment provided by the Gym library.

- "max_episode_timesteps": 100: This parameter sets the maximum number of timesteps allowed in a single episode of the reinforcement learning task. An episode represents a complete interaction between the agent and the environment, and timesteps are individual time steps within that episode. Setting a maximum number of timesteps can be useful to limit the duration of an episode, especially in cases where the environment does not naturally terminate.

Replay mechanism:

```
{
   "type": "replay"
}
```

- "type": "replay": This line indicates the type of mechanism being used, and in this case, it's labelled as "replay."

In the context of deep reinforcement learning, experience replay involves storing past experiences (tuples of state, action, reward, and next state) in a replay buffer. During training, random batches of experiences are sampled from this buffer and used to update the neural network. Experience replay helps break the temporal correlation in the sequence of experiences, making the training process more stable and efficient.

The policy:

```
{
    "agent": "ppo",
    "network": {"type": "auto", "internal_rnn": false},
    "batch_size": 4,
    "update_frequency": 4,
    "learning_rate": 0.001,
    "subsampling_fraction": 0.3,
    "optimization_steps": 10,
    "likelihood_ratio_clipping": 0.1,
    "discount": 0.9,
    "critic_network": {"type": "auto", "internal_rnn": false},
    "critic_optimizer": 1.0,
    "preprocessing": null,
    "exploration": 0.0,
    "variable_noise": 0.0,
    "l2_regularization": 0.0,
    "entropy_regularization": 0.001
}
```

Figure 20: Modelled policy [6]

- "agent": "ppo": Indicates that the reinforcement learning algorithm being configured is Proximal Policy Optimization (PPO). PPO is a policy optimization algorithm commonly used for training agents in reinforcement learning.

- "network": {"type": "auto", "internal_rnn": false}: Specifies the neural network architecture for the policy. In this case, it's set to "auto," indicating that the network type is automatically determined. The internal_rnn parameter is set to false, suggesting that there is no internal recurrent neural network (RNN) used in the policy network.
- "batch_size": 4: Sets the size of the batches used during training to 4 samples.
- "update_frequency": 4: Defines how often the policy should be updated. In this case, it's set to every 4 batches.
- "learning_rate": 0.001: Specifies the learning rate used during optimization.
- "subsampling_fraction": 0.3: Determines the fraction of the collected data used for training. In this case, 30% of the collected data is subsampled for training.
- "optimization_steps": 10: Sets the number of optimization steps to take per update.
- "likelihood_ratio_clipping": 0.1: Introduces a constraint on the ratio of new and old policy probabilities to prevent large policy updates.
- "discount": 0.9: Defines the discount factor for future rewards in the reinforcement learning problem.
- "critic_network": {"type": "auto", "internal_rnn": false}: Specifies the neural network architecture for the critic (value function). Similar to the policy network, it's set to "auto" with no internal RNN.
- "critic_optimizer": 1.0: Specifies the coefficient for the critic loss in the overall objective function.
- "preprocessing": null: Indicates that no specific preprocessing is applied to the input data.
- "exploration": 0.0: Sets the exploration parameter to 0.0, suggesting that there is no explicit exploration strategy.
- "variable_noise": 0.0: Specifies the amount of noise to add to the policy parameters during training.
- "l2_regularization": 0.0: Specifies the L2 regularization strength.
- "entropy_regularization": 0.001: Introduces regularization on the entropy of the policy distribution.

The RL is based on the Tensorforce library and allows the combination of a variety of popular deep reinforcement learning models. Further details are found in the Tensorforce documentation shown in Figure 22. Problem-specific configurations for the order dispatching task are the following (initialize_env.py), that are available in [6] in the production/envs/ section:

- State representation, i.e. which information elements are part of the state vector
- Reward function (incl. consideration of multiple objective functions and weighted reward functions according to action subset type)
- Action representation, i.e. which actions are allowed (e.g., "idling" action) and type of mapping of discrete action number to dispatching decisions
- Episode definition and limit
- RL-specific parameters such as learning rate, discount rate, neural network configuration etc. are defined in the Tensorforce agent configuration represented in Figure 18.

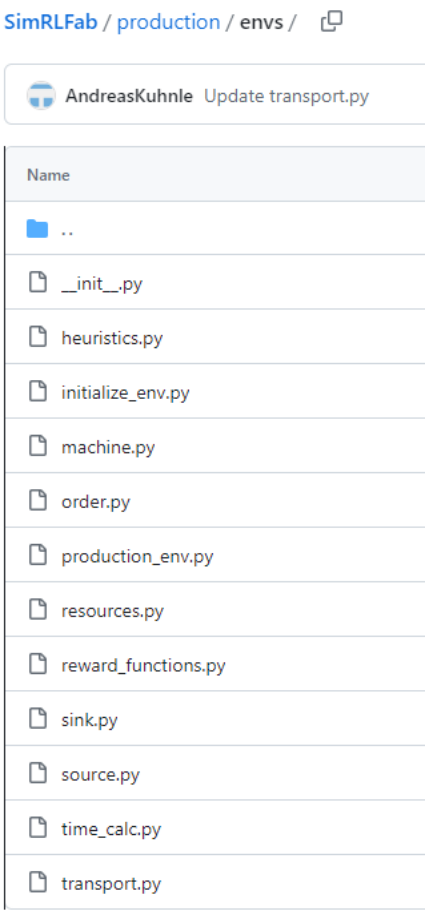

Figure 21: Problem-specific configurations for the order dispatching task [6]

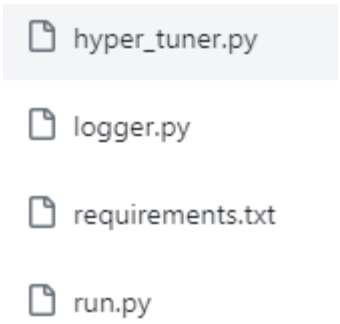

Figure 22: Tensorforce library used in the work [6]

Extentions not yet implemented (future work):

- job due dates
- Batch processing
- Alternative maintenance strategies (predictive, etc.)
- Alternative strategies for order sequencing and processing at machines
- Mutliple RL-agents for several production control tasks

9. Simplified algorithm development

The final contribution of this work is the development of a simplified algorithm which simulates the use of RL in a manufacturing problem. The system is composed of 3 machines and 2 robots who works as orders' dispatchers. The algorithm is used to train the robots to decide how to dispacth orders based on their due date, processing time and buffer level of the machines.

50 orders are considered to run the algorithm which are generated randomically. Cumulative reward is based on completion time and due date. A Q-Learning algorithm is used and the specific variables are set to, considering the values that are moslty used:

- learning_rate=0.1
- discount_factor=0.9
- exploration_prob=0.1

The algorithm is run for 500 episodes and the results are plotte in a graph.

As we can see the algorithm converge immediately, these is due to the simplicity of the case study, but it shows the potential and effectiveness in manufacturing problems as the simulated one.

```
import gymnasium as gym
import numpy as np
import matplotlib.pyplot as plt

class OrderDispatchEnvironment(gym.Env):
    def __init__(self, num_jobs=50, num_machines=3, num_robots=2):
        self.num_jobs = num_jobs
        self.num_machines = num_machines
        self.num_robots = num_robots
        self.current_job = 0
        self.machine_availability = np.ones(num_machines, dtype=int)
        self.robot_availability = np.ones(num_robots, dtype=int)

        self.observation_space = gym.spaces.Tuple((
            gym.spaces.Discrete(num_jobs),  # Current job
            gym.spaces.MultiBinary(num_machines),  # Machines availability
            gym.spaces.MultiBinary(num_robots)  # Robots availability
        ))
        self.action_space = gym.spaces.Discrete(num_machines)  # Choose a
machine to dispatch the order

    def reset(self):
        self.current_job = 0
        self.machine_availability = np.ones(self.num_machines, dtype=int)
        self.robot_availability = np.ones(self.num_robots, dtype=int)
        return self._get_state()

    def step(self, action):
        if self.current_job < self.num_jobs:
            # Simulate the execution of the order
```

```
            machine_selected = action
            completion_time = 1  # Simulated completion time for the order
            self.machine_availability[machine_selected] = 0  # Mark the machine
as busy

            # Update the state
            self.current_job += 1
            state = self._get_state()

            # Calculate reward based on completion time and due date
            reward = self.calculate_reward(completion_time, due_date=1)  #
Replace '1' with the actual due date

            done = (self.current_job == self.num_jobs)

            return state, reward, done, {}
        else:
            # End of jobs
            return self._get_state(), 0, True, {}

    def calculate_reward(self, completion_time, due_date):
        efficient_utilization_reward = 0.1
        late_penalty = 0.5

        if completion_time <= due_date:
            reward = efficient_utilization_reward - late_penalty * (due_date -
completion_time)
        else:
            reward = -late_penalty * (completion_time - due_date)

        return reward

    def _get_state(self):
        return (self.current_job, self.machine_availability.copy(),
self.robot_availability.copy())

class QLearningAgent:
    def __init__(self, state_space, action_space, learning_rate=0.1,
discount_factor=0.9, exploration_prob=0.1):
        self.learning_rate = learning_rate
        self.discount_factor = discount_factor
        self.exploration_prob = exploration_prob
        self.state_space = state_space
        self.action_space = action_space
        self.q_table = np.zeros((state_space[0].n, 2 ** state_space[1].n, 2 **
state_space[2].n, action_space.n))

    def choose_action(self, state):
        if np.random.rand() < self.exploration_prob:
            return np.random.choice(self.action_space.n)
        else:
            state_index = state[0]
            combined_state = np.concatenate((state[1], state[2]))
            return np.argmax(self.q_table[state_index, combined_state, :])
```

```
    def update_q_table(self, state, action, reward, next_state):
        combined_state = np.concatenate((state[1], state[2]))
        combined_next_state = np.concatenate((next_state[1], next_state[2]))
        combined_next_state_index = int(''.join(map(str, combined_next_state)),
2) % self.q_table.shape[1]

        # Verifica delle dimensioni dell'array
        print("Dimensioni di self.q_table:", self.q_table.shape)

        # Verifica dei valori prima di accedere all'array
        print("Valori di next_state[0] e combined_next_state:", next_state[0],
combined_next_state)

        # Converti l'array binario in un indice intero con clamp alla
dimensione massima
        combined_state_index = int(''.join(map(str, combined_state)), 2) %
self.q_table.shape[1]
        best_next_action = np.argmax(self.q_table[state[0],
combined_state_index, :])

        try:
            print("Before update - Q value:", self.q_table[state[0],
combined_state_index, action])

            # Controlla le dimensioni prima di aggiornare il valore
            if (next_state[0] % self.q_table.shape[0] < self.q_table.shape[0]
and
                    combined_next_state_index < self.q_table.shape[2] and
                    best_next_action < self.q_table.shape[3]):
                self.q_table[state[0], combined_state_index, action] +=
self.learning_rate * \
                    (reward + self.discount_factor * self.q_table[next_state[0]
% self.q_table.shape[0], combined_next_state_index, best_next_action] -
self.q_table[state[0], combined_state_index, action])
            else:
                print("Warning: Indici fuori dai limiti durante
l'aggiornamento")

            print("After update - Q value:", self.q_table[state[0],
combined_state_index, action])

        except IndexError as e:
            print(f"Error: {e}")
            print(f"State: {state}")
            print(f"Action: {action}")
            print(f"Reward: {reward}")
            print(f"Next State: {next_state}")
            print(f"Combined State Index: {combined_state_index}")
            print(f"Combined Next State Index: {combined_next_state_index}")
            print(f"Best Next Action: {best_next_action}")
            print(f"Q-table shape: {self.q_table.shape}")

            # Re-raise the exception to terminate the program
            raise
```

```
# Hyperparameters
num_episodes = 500
num_jobs = 50

# Instantiate the environment and agent
env = OrderDispatchEnvironment(num_jobs=num_jobs, num_machines=3, num_robots=2)
agent = QLearningAgent(state_space=env.observation_space,
action_space=env.action_space)

# Lists to store results for plotting
total_rewards = []

# Training loop
for episode in range(num_episodes):
    state = env.reset()
    total_reward = 0
    done = False

    while not done:
        action = agent.choose_action(state)
        next_state, reward, done, _ = env.step(action)

        agent.update_q_table(state, action, reward, next_state)

        state = next_state
        total_reward += reward

    total_rewards.append(total_reward)
    print(f"Episode: {episode + 1}, Total Reward: {total_reward}")

# Plotting the results
plt.plot(total_rewards)
plt.xlabel('Episode')
plt.ylabel('Total Reward')
plt.title('Training Progress')
plt.show()

# Evaluate the learned policy
test_episodes = 10
test_rewards = []

for _ in range(test_episodes):
    state = env.reset()
    total_reward = 0
    done = False

    while not done:
        action = agent.choose_action(state)
        next_state, reward, done, _ = env.step(action)
        state = next_state
        total_reward += reward

    test_rewards.append(total_reward)
```

```
# Print average reward during testing
average_test_reward = np.mean(test_rewards)
print(f"Average Test Reward: {average_test_reward}")
```

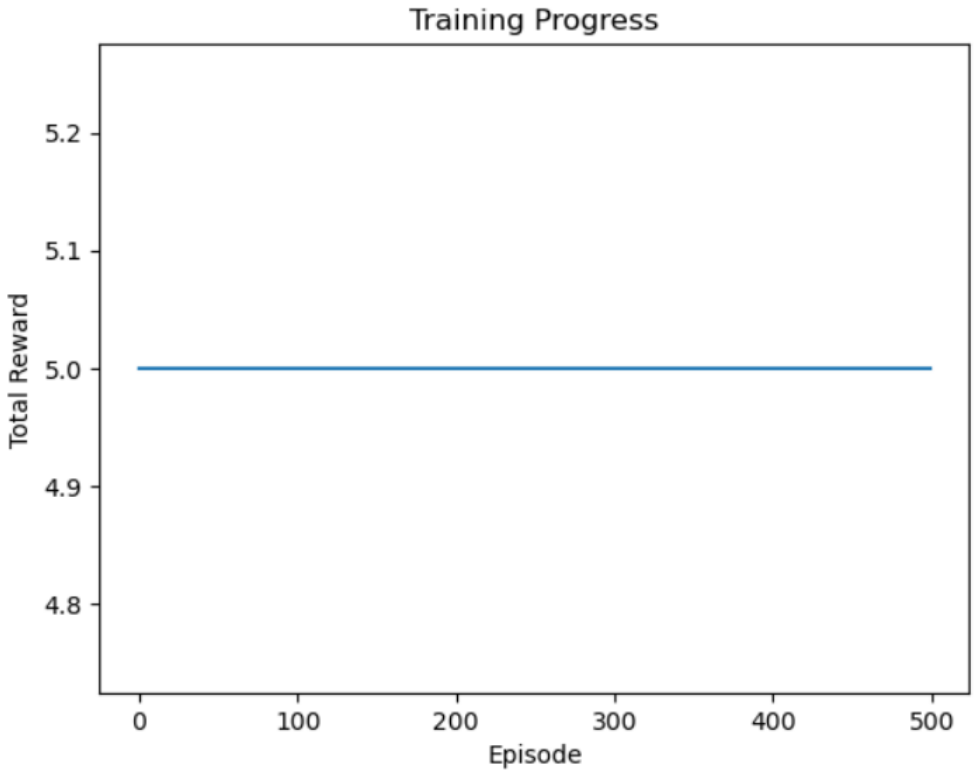

Figure 22: Representation of total reward and convergence of the developed algorithm

10. Conclusion

In conclusion, the exploration of Reinforcement Learning (RL) applications in manufacturing has not only revealed its transformative potential but has also highlighted practical advantages when compared to traditional heuristic methods and other machine learning approaches. The papers studied throughout this study underscore the adaptability and effectiveness of RL algorithms in optimizing resource utilization, enhancing decision-making, and improving overall system efficiency within manufacturing contexts.

Table 2 shows the result of the reseach, in particular in which area RL is used and its percentage.As we can see it is mostly applied to scheduling, maintence and process control problems compared to autonomous and sustainable manufacturing and motion planning. Futhure work will focus on developing algorithms that are able to interconnect different goals to be able to optimize real production processes as a whole as the surge in computing power accessibility, coupled with the exponential growth of data from sensors and IoT devices in manufacturing systems, has created a fertile ground for RL applications. This aligns seamlessly with the Industry 4.0 paradigm, emphasizing the integration of digital technologies into manufacturing. RL's adaptability to dynamic environments and its capacity to continuously learn and optimize processes over time make it well-suited for the evolving needs of the manufacturing industry

Area of Application	Number of papers	%
Autonomous Manufacturing	7	7%
Sustainable Manufacturing	3	3%
Maintence Strategies and Quality	16	16%
Motion Planning	4	4%
Process control	14	14%
Scheduling	54	55%
total	98	100%

Table 2: Thesis findings

The state of the art in RL for manufacturing is poised to redefine how industries approach complex problem-solving. The integration of RL into manufacturing processes reflects a paradigm shift, as intelligent agents learn and adapt in real-time, contributing to agile and responsive systems. The advantages of RL, such as improved adaptability, reduced reliance on explicit programming, and the capacity to handle non-linear and dynamic relationships, position it as a compelling choice for addressing the intricate challenges of modern manufacturing.

Unlike conventional heuristic methods that often rely on predetermined rules, RL offers a dynamic and learning-driven approach. The ability of RL agents to adapt to changing conditions and learn optimal strategies over time presents a distinct advantage in complex manufacturing environments.

This adaptability is particularly evident in scenarios where the system dynamics evolve or face uncertainties.

Furthermore, in contrast to some traditional machine learning methods that might require extensive labelled datasets, RL's ability to learn from interaction and experience proves advantageous in situations where data availability is a challenge. This feature becomes especially pertinent in manufacturing settings where acquiring labelled data for every conceivable scenario can be impractical.

However, it's important to acknowledge the practical challenges in implementing RL algorithms. Designing and deploying RL algorithms for manufacturing processes involve intricate considerations. While simulations offer a controlled environment for algorithm development and testing, translating these algorithms into real-world applications introduces a set of challenges. Practical issues such as hardware constraints, sensor integration, and real-time responsiveness must be addressed for seamless deployment on the shop floor.

Moreover, the complexity of RL algorithms raises challenges in algorithmic design and implementation. The fine-tuning of hyperparameters, ensuring convergence, and managing the computational demands of sophisticated RL models pose practical challenges. Balancing the need for a high level of accuracy with real-time responsiveness remains a delicate trade-off that practitioners must navigate.

Despite these challenges, the potential benefits of RL in manufacturing are substantial. Its adaptive nature allows for continuous learning and optimization, offering a promising avenue for addressing dynamic manufacturing environments. While RL algorithms are predominantly simulated during development, their successful implementation in real-world scenarios heralds a new era in smart manufacturing.

As we look ahead, collaborative efforts between researchers, industry professionals, and policymakers become crucial. The continued exploration of RL's practical challenges and iterative refinement of algorithms will pave the way for increased real-world adoption. Striking a balance between algorithmic sophistication and practical implementation will be key in realizing the full potential of RL in reshaping the landscape of modern manufacturing.

Nevertheless, challenges persist, and the road ahead involves a collaborative effort between academia and industry. As RL technologies mature, addressing scalability, interpretability, and ethical considerations will be crucial. However, the practical advantages observed in this study, particularly in comparison to heuristics methods and other machine learning approaches, underscore RL's potential as a transformative force in shaping the future of manufacturing.

11. Acknowledgements

As I reflect on the completion of this significant academic milestone, my heart is filled with profound gratitude for the multitude of individuals who have played an important role in this journey.

I want to thank myself for the commitment, resilience, and dedication that have led to the successful completion of this amazing but challenging journey. These years have showed me what perseverance and hard work are and I bet that this will help me to achieve future goals.

To my family, whose unwavering support has been my anchor throughout this academic journey, I am sincerely grateful. Your belief in my abilities, the sacrifices made, and the constant encouragement have been one of the driving forces behind my success. This accomplishment is as much yours as it is mine.

A sincere thank you to my colleagues for the shared insights and collaborative spirit, which have made this academic pursuit to remember and fun.

I would also like to express my sincere gratitude to my advisor, Giulia Bruno, for the invaluable guidance provided during the internship in Barilla and thesis writing process. Your understanding of my specific circumstances and your unwavering support have been indispensable, shaping the quality of this work and giving me the possibility to work while finishing my academic career.

A special chapter of my academic journey was the opportunity granted by Polytechnic di Torino to embark on a six-month Erasmus program in Mexico. I extend my deepest gratitude to the university for providing me with this enriching experience. To all the individuals I had the privilege of meeting during my time in Mexico, thank you for contributing to an experience that was not only academically rewarding but also culturally and personally transformative. The friendships forged and the lessons learned during that period have added a unique dimension to my personal and academic growth.

Furthermore, I extend a particular thank you to my mother, whose unwavering trust and support have been my guiding light. With the backing of my entire family, including my grandparents who raised me and thought me sound principles, she enabled me not only to pursue my studies in Turin but also to embark on unique and formative experiences, such as a year in the United States and six months in Mexico.

A special thanks to my academic buddy, Alessia, whom I have known from the beginning of this journey. We connected instantly, supporting each other through thick and thin, and sharing most memories and adventures in Turin. Alessia, your friendship has been a cherished aspect of this experience, and I am grateful for the bond we've formed, and I know it is only going to grow.

I would also like to thank you my cousin Claudio for helping me choosing Polytechnic di Torino and moving to this new city, my friend Valeria for the amazing Pizza dinners that will always be part of my Turin memories. Finally, my aunt and uncle from Parma, who supported me and kept me company every weekend during the last year since starting to work and studying during the weekend.

As I transition to exciting new adventure and challenges, I carry with me the collective support, lessons learned, and the gratitude for the incredible individuals who have been pivotal in shaping this chapter of my academic and personal journey.

I know that’s not an end but the start of a wonderful new chapter rich of new challenges, personal and business achievements.

Thank you.
Warm regards,

Sonia

12. References

[1] A. Acernese, A. Yerudkar and C. Del Vecchio, "A Novel Reinforcement Learning-based Unsupervised Fault Detection for Industrial Manufacturing Systems," 2022 American Control Conference (ACC), Atlanta, GA, USA, 2022, pp. 2650-2655, doi: 10.23919/ACC53348.2022.9867763.

[2] A. G. Dharmawan, Y. Xiong, S. Foong and G. Song Soh, "A Model-Based Reinforcement Learning and Correction Framework for Process Control of Robotic Wire Arc Additive Manufacturing," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 4030-4036, doi: 10.1109/ICRA40945.2020.9197222.

[3] A. S. Xanthopoulos, A. Kiatipis, D. E. Koulouriotis and S. Stieger, "Reinforcement Learning-Based and Parametric Production-Maintenance Control Policies for a Deteriorating Manufacturing System," in IEEE Access, vol. 6, pp. 576-588, 2018, doi: 10.1109/ACCESS.2017.2771827.

[4] Achamrah, F. E., & Attajer, A. (2023). Multi-objective reinforcement learning-based framework for solving selective maintenance problems in reconfigurable cyber-physical manufacturing systems. International Journal of Production Research, 1–23. https://doi.org/10.1080/00207543.2023.2240433

[5] Altenmüller, T., Stüker, T., Waschneck, B., Kuhnle, A., & Lanza, G. (2020). Reinforcement learning for an intelligent and autonomous production control of complex job-shops under time constraints. Production Engineering, 14(3), 319–328. https://doi.org/10.1007/s11740-020-00967-8

[6] AndreasKuhnle. (n.d.). SimRLFab/README.md at master · AndreasKuhnle/SimRLFab. GitHub. https://github.com/AndreasKuhnle/SimRLFab

[7] B. Waschneck et al., "Deep reinforcement learning for semiconductor production scheduling," 2018 29th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC), Saratoga Springs, NY, USA, 2018, pp. 301-306, doi: 10.1109/ASMC.2018.8373191.

[8] Chen, T., Sampath, V., May, M. C., Shan, S., Jorg, O., Martín, J. J. A., Stamer, F., Fantoni, G., Tosello, G., & Calaon, M. (2023). Machine Learning in Manufacturing towards Industry 4.0: From 'For Now' to 'Four-Know.' Applied Sciences, 13(3), 1903.

[9] Chen, Z., Zhang, L., Wang, X., & Wang, K. (2023). Cloud–edge collaboration task scheduling in cloud manufacturing: An attention-based deep reinforcement learning approach. Computers & Industrial Engineering, 177, 109053. https://doi.org/10.1016/j.cie.2023.109053

[10] Chua, PC, Moon, SK, Ng, YT, Ng, HY, & Lopez, M. "Discovery of Customized Dispatching Rule for Single-Machine Production Scheduling Using Deep Reinforcement Learning." Proceedings of the ASME 2022 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 2: 42nd Computers and Information in Engineering Conference (CIE). St. Louis, Missouri, USA. August 14–17, 2022. V002T02A072. ASME. https://doi.org/10.1115/DETC2022-89829

[11] Chung, J., Shen, B., Law, A. W., & Kong, Z. (2022). Reinforcement learning-based defect mitigation for quality assurance of additive manufacturing. Journal of Manufacturing Systems, 65, 822–835. https://doi.org/10.1016/j.jmsy.2022.11.008

[12] D. Johnson, G. Chen and Y. Lu, "Multi-Agent Reinforcement Learning for Real-Time Dynamic Production Scheduling in a Robot Assembly Cell," in IEEE Robotics and Automation Letters, vol. 7, no. 3, pp. 7684-7691, July 2022, doi: 10.1109/LRA.2022.3184795.

[13] Del Real Torres, A., Andreiana, D. S., Roldán, Á. O., Bustos, A. H., & Acevedo, L. (2022d). A Review of Deep Reinforcement Learning Approaches for Smart Manufacturing in Industry 4.0 and 5.0 framework. Applied Sciences, 12(23), 12377.

[14] Dharmadhikari, S., Menon, N., & Basak, A. (2023). A reinforcement learning approach for process parameter optimization in additive manufacturing. Additive Manufacturing, 71, 103556. https://doi.org/10.1016/j.addma.2023.103556

[15] Dornheim, J., Link, N., & Gumbsch, P. (2019). Model-free adaptive optimal control of episodic fixed-horizon manufacturing processes using reinforcement learning. International Journal of Control, Automation and Systems, 18(6), 1593–1604. https://doi.org/10.1007/s12555-019-0120-7

[16] F. Yang, T. Feng, F. Xu, H. Jiang and C. Zhao, "Collaborative clustering parallel reinforcement learning for edge-cloud digital twins manufacturing system," in China Communications, vol. 19, no. 8, pp. 138-148, Aug. 2022, doi: 10.23919/JCC.2022.08.011.

[17] Fan, F., Xu, G., Feng, N., Li, L., Jiang, W., Yu, L., & Xiong, X. (2023). Spatiotemporal path tracking via deep reinforcement learning of robot for manufacturing internal logistics. Journal of Manufacturing Systems, 69, 150–169. https://doi.org/10.1016/j.jmsy.2023.06.011

[18] Gu, W., Li, Y., Tang, D., Wang, X., & Yuan, M. (2022). Using real-time manufacturing data to schedule a smart factory via reinforcement learning. Computers & Industrial Engineering, 171, 108406. https://doi.org/10.1016/j.cie.2022.108406

[19] H. Rasay, F. Azizi, M. Salmani and F. Naderkhani, "A Reinforcement Learning Algorithm for Optimal Dynamic Policies of Joint Condition-based Maintenance and Condition-based Production," 2023 IEEE International Conference on Prognostics and Health Management (ICPHM), Montreal, QC, Canada, 2023, pp. 134-138, doi: 10.1109/ICPHM57936.2023.10193968.

[20] H. Rasay, F. Naderkhani and A. M. Golmohammadi, "Reinforcement Learning based on Stochastic Dynamic Programming for Condition-based Maintenance of Deteriorating Production Processes," 2022 IEEE International Conference on Prognostics and Health Management (ICPHM), Detroit (Romulus), MI, USA, 2022, pp. 17-24, doi: 10.1109/ICPHM53196.2022.9815668.

[21] Hameed, M. S. A., & Schwung, A. (2023). Graph neural networks-based scheduler for production planning problems using reinforcement learning. Journal of Manufacturing Systems, 69, 91–102. https://doi.org/10.1016/j.jmsy.2023.06.005

[22] He, Z., Tran, K. P., Thomassey, S., Zeng, X., Xu, J., & Yi, C. (2022). Multi-objective optimization of the textile manufacturing process using deep-Q-network based multi-agent reinforcement learning. Journal of Manufacturing Systems, 62, 939–949. https://doi.org/10.1016/j.jmsy.2021.03.017

[23] Hu, L., Liu, Z., Hu, W., Wang, Y., Tan, J., & Wu, F. (2020). Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network. Journal of Manufacturing Systems, 55, 1–14. https://doi.org/10.1016/j.jmsy.2020.02.004

[24] Huang, J., Chang, Q., & Arinez, J. (2020). Deep reinforcement learning based preventive maintenance policy for serial production lines. Expert Systems With Applications, 160, 113701. https://doi.org/10.1016/j.eswa.2020.113701

[25] Huang, J., Su, J., & Chang, Q. (2022). Graph neural network and multi-agent reinforcement learning for machine-process-system integrated control to optimize production

yield. Journal of Manufacturing Systems, 64, 81–93. https://doi.org/10.1016/j.jmsy.2022.05.018

[26] I. -B. Park, J. Huh, J. Kim and J. Park, "A Reinforcement Learning Approach to Robust Scheduling of Semiconductor Manufacturing Facilities," in IEEE Transactions on Automation Science and Engineering, vol. 17, no. 3, pp. 1420-1431, July 2020, doi: 10.1109/TASE.2019.2956762.

[27] J. Giner, R. Lamprecht, V. Gallina, C. Laflamme, L. Sielaff and W. Sihn, "Demonstrating Reinforcement Learning for Maintenance Scheduling in a Production Environment," 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vasteras, Sweden, 2021, pp. 1-8, doi: 10.1109/ETFA45728.2021.9613205.

[28] Jeon, S., Lee, D., Park, K. T., Park, K., Noh, S. D., & Arinez, J. (2022). Design and Implementation of Simulation-Based Scheduling System with Reinforcement Learning for Re-Entrant Production Lines. Machines, 10(12), 1169. https://doi.org/10.3390/machines10121169

[29] Jeong, Y., Agrawal, T. K., Flores-García, E., & Wiktorsson, M. (2021). A reinforcement learning model for material handling task assignment and route planning in dynamic production logistics environment. Procedia CIRP, 104, 1807–1812. https://doi.org/10.1016/j.procir.2021.11.305

[30] Joo, T., Jun, H., & Shin, D. (2022). Task allocation in Human–Machine manufacturing systems using deep reinforcement learning. Sustainability, 14(4), 2245. https://doi.org/10.3390/su14042245

[31] Karigiannis, J., Laurin, P., Liu, S., Holovashchenko, V., Lizotte, A., Roux, V. L., & Boulet, P. (2022). Reinforcement learning enabled Self-Homing of industrial robotic manipulators in manufacturing. Manufacturing Letters, 33, 909–918. https://doi.org/10.1016/j.mfglet.2022.07.111

[32] Kim, S., Kim, D. D., & Anthony, B. (2019). Dynamic Control of a Fiber Manufacturing Process using Deep Reinforcement Learning. ResearchGate. https://www.researchgate.net/publication/337532155_Dynamic_Control_of_a_Fiber_Manufacturing_Process_using_Deep_Reinforcement_Learning

[33] Kim, T., Kim, Y., Lee, D., & Kim, M. (2022). Reinforcement learning approach to scheduling of precast concrete production. Journal of Cleaner Production, 336, 130419. https://doi.org/10.1016/j.jclepro.2022.130419

[34] Kim, Y. G., Lee, S., Son, J., Bae, H., & Chung, B. D. (2020). Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system. Journal of Manufacturing Systems, 57, 440–450. https://doi.org/10.1016/j.jmsy.2020.11.004

[35] Kuhnle, A., Kaiser, J., Theiß, F., Stricker, N., & Lanza, G. (2020). Designing an adaptive production control system using reinforcement learning. Journal of Intelligent Manufacturing, 32(3), 855–876. https://doi.org/10.1007/s10845-020-01612-y

[36] Kuhnle, A., May, M. C., Schäfer, L., & Lanza, G. (2021). Explainable reinforcement learning in production control of job shop manufacturing system. International Journal of Production Research, 60(19), 5812–5834. https://doi.org/10.1080/00207543.2021.1972179

[37] L. Zhang, C. Yang, Y. Yan and Y. Hu, "Distributed Real-Time Scheduling in Cloud Manufacturing by Deep Reinforcement Learning," in IEEE Transactions on Industrial Informatics, vol. 18, no. 12, pp. 8999-9007, Dec. 2022, doi: 10.1109/TII.2022.3178410.

[38] L. Zhang, C. Yang, Y. Yan and Y. Hu, "Distributed Real-Time Scheduling in Cloud Manufacturing by Deep Reinforcement Learning," in IEEE Transactions on Industrial Informatics, vol. 18, no. 12, pp. 8999-9007, Dec. 2022, doi: 10.1109/TII.2022.3178410.

[39] Lang, S., Kuetgens, M., Reichardt, P., & Reggelin, T. (2021). Modeling Production Scheduling Problems as Reinforcement Learning Environments based on Discrete-Event Simulation and OpenAI Gym. IFAC-PapersOnLine, 54(1), 793–798. https://doi.org/10.1016/j.ifacol.2021.08.093

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

[41] Lei, J., Jing, H., Chang, F., Dassari, S., & Ding, K. (2022). Reinforcement learning-based dynamic production-logistics-integrated tasks allocation in smart factories. International Journal of Production Research, 61(13), 4419–4436. https://doi.org/10.1080/00207543.2022.2142314

[42] Leng, J., Wang, X., Wu, S., Jin, C., Tang, M., Li, R., Vogl, A., & Liu, H. (2022). A multi-objective reinforcement learning approach for resequencing scheduling problems in automotive manufacturing systems. International Journal of Production Research, 61(15), 5156–5175. https://doi.org/10.1080/00207543.2022.2098871

[43] Li, C., & Chang, Q. (2022). Hybrid feedback and reinforcement learning-based control of machine cycle time for a multi-stage production system. Journal of Manufacturing Systems, 65, 351–361. https://doi.org/10.1016/j.jmsy.2022.09.020

[44] Li, J., Pang, D., Zheng, Y., Guan, X., & Le, X. (2022). A flexible manufacturing assembly system with deep reinforcement learning. Control Engineering Practice, 118, 104957. https://doi.org/10.1016/j.conengprac.2021.104957

[45] Li, Y., Du, J., & Jiang, W. (2023). Reinforcement learning for process control with application in semiconductor manufacturing. IISE Transactions, 1–15. https://doi.org/10.1080/24725854.2023.2219290

[46] Liang, H., Wen, X., Liu, Y., Zhang, H., Zhang, L., & Wang, L. (2021). Logistics-involved QoS-aware service composition in cloud manufacturing with deep reinforcement learning. Robotics and Computer-Integrated Manufacturing, 67, 101991. https://doi.org/10.1016/j.rcim.2020.101991

[47] Liu, J., Qiao, F., Zou, M., Zinn, J., Ma, Y., & Vogel-Heuser, B. (2022). Dynamic scheduling for semiconductor manufacturing systems with uncertainties using convolutional neural networks and reinforcement learning. Complex & Intelligent Systems, 8(6), 4641–4662. https://doi.org/10.1007/s40747-022-00844-0

[48] Liu, J., Qiao, F., Zou, M., Zinn, J., Ma, Y., & Vogel-Heuser, B. (2022b). Dynamic scheduling for semiconductor manufacturing systems with uncertainties using convolutional neural networks and reinforcement learning. Complex & Intelligent Systems, 8(6), 4641–4662. https://doi.org/10.1007/s40747-022-00844-0

[49] Liu, Y., Ping, Y., Zhang, L., Wang, L., & Xu, X. (2023). Scheduling of decentralized robot services in cloud manufacturing with deep reinforcement learning. Robotics and Computer-Integrated Manufacturing, 80, 102454. https://doi.org/10.1016/j.rcim.2022.102454

[50] M. F. Alam, M. Shtein, K. Barton and D. Hoelzle, "Reinforcement Learning Enabled Autonomous Manufacturing Using Transfer Learning and Probabilistic Reward Modeling," in IEEE Control Systems Letters, vol. 7, pp. 508-513, 2023, doi: 10.1109/LCSYS.2022.3188014.

[51] M. Feng and Y. Li, "Predictive Maintenance Decision Making Based on Reinforcement Learning in Multistage Production Systems," in IEEE Access, vol. 10, pp. 18910-18921, 2022, doi: 10.1109/ACCESS.2022.3151170.

[52] M. Salmani, F. Azizi, H. Rasay and F. Naderkhani, "Dynamic Maintenance for a Large Scale Identical Parallel Manufacturing Systems Using Reinforcement Learning," 2023 Annual Reliability and Maintainability Symposium (RAMS), Orlando, FL, USA, 2023, pp. 1-8, doi: 10.1109/RAMS51473.2023.10088200.

[53] Marchesano, M. G., Guizzi, G., Santillo, L. C., & Vespoli, S. (2021). A Deep Reinforcement Learning approach for the throughput control of a Flow-Shop production system. IFAC-PapersOnLine, 54(1), 61–66. https://doi.org/10.1016/j.ifacol.2021.08.006

[54] Mayer, S., Classen, T., & Endisch, C. (2021). Modular production control using deep reinforcement learning: proximal policy optimization. Journal of Intelligent Manufacturing, 32(8), 2335–2351. https://doi.org/10.1007/s10845-021-01778-z

[55] Mueller-Zhang, Z., Antonino, P. O., & Kuhn, T. S. (2021). Integrated Planning and Scheduling for Customized Production using Digital Twins and Reinforcement Learning. IFAC-PapersOnLine, 54(1), 408–413. https://doi.org/10.1016/j.ifacol.2021.08.046

[56] N. Krippendorff and C. Schwindt, "Control of Shared Production Buffers: A Reinforcement Learning Approach," 2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, Singapore, 2021, pp. 703-707, doi: 10.1109/IEEM50564.2021.9673034.

[57] Nian, R., Liu, J., & Huang, B. (2020). A review On reinforcement learning: Introduction and applications in industrial process control. Computers & Chemical Engineering, 139, 106886.

[58] O. M. Manyar, Z. McNulty, S. Nikolaidis and S. K. Gupta, "Inverse Reinforcement Learning Framework for Transferring Task Sequencing Policies from Humans to Robots in Manufacturing Applications," 2023 IEEE International Conference on Robotics and Automation (ICRA), London, United Kingdom, 2023, pp. 849-856, doi: 10.1109/ICRA48891.2023.10160687.

[59] Oliff, H., Liu, Y., Kumar, M., Williams, M. D., & Ryan, M. J. (2020). Reinforcement learning for facilitating human-robot-interaction in manufacturing. Journal of Manufacturing Systems, 56, 326–340. https://doi.org/10.1016/j.jmsy.2020.06.018

[60] Overbeck, L., Hugues, A., May, M. C., Kuhnle, A., & Lanza, G. (2021). Reinforcement learning based production control of semi-automated manufacturing systems. Procedia CIRP, 103, 170–175. https://doi.org/10.1016/j.procir.2021.10.027

[61] Pahwa, D., & Starly, B. (2021). Dynamic matching with deep reinforcement learning for a two-sided Manufacturing-as-a-Service (MaaS) marketplace. Manufacturing Letters, 29, 11–14. https://doi.org/10.1016/j.mfglet.2021.05.005

[62] Paraschos, P. D., Koulinas, G. K., & Koulouriotis, D. E. (2020). Reinforcement learning for combined production-maintenance and quality control of a manufacturing system with deterioration failures. Journal of Manufacturing Systems, 56, 470–483. https://doi.org/10.1016/j.jmsy.2020.07.004

[63] Paraschos, P. D., Koulinas, G. K., & Koulouriotis, D. E. (2023). A reinforcement learning/ad-hoc planning and scheduling mechanism for flexible and sustainable manufacturing systems. Flexible Services and Manufacturing Journal. https://doi.org/10.1007/s10696-023-09496-9

[64] Park, K. T., Son, Y. H., Ko, S. W., & Noh, S. D. (2021). Digital twin and reinforcement Learning-Based Resilient Production control for micro smart factory. Applied Sciences, 11(7), 2977. https://doi.org/10.3390/app11072977

[65] Petrik, J., & Bambach, M. (2023). Reinforcement learning and optimization based path planning for thin-walled structures in wire arc additive manufacturing. Journal of Manufacturing Processes, 93, 75–89. https://doi.org/10.1016/j.jmapro.2023.03.013

[66] Ping, Y., Liu, Y., Zhang, L., Wang, L., & Xu, X. (2023). Sequence generation for multi-task scheduling in cloud manufacturing with deep reinforcement learning. Journal of Manufacturing Systems, 67, 315–337. https://doi.org/10.1016/j.jmsy.2023.02.009

[67] Pires, F., Leitão, P., Moreira, A. P., & Ahmad, B. (2023). Reinforcement learning based trustworthy recommendation model for digital twin-driven decision-support in manufacturing systems. Computers in Industry, 148, 103884. https://doi.org/10.1016/j.compind.2023.103884

[68] Popper, J., Yfantis, V., & Ruskowski, M. (2021). Simultaneous Production and AGV Scheduling using Multi-Agent Deep Reinforcement Learning. Procedia CIRP, 104, 1523–1528. https://doi.org/10.1016/j.procir.2021.11.257

[69] Qin, Z., Johnson, D., & Lu, Y. (2023). Dynamic production scheduling towards self-organizing mass personalization: A multi-agent dueling deep reinforcement learning approach. Journal of Manufacturing Systems, 68, 242–257. https://doi.org/10.1016/j.jmsy.2023.03.003

[70] Qu, S., Wang, J., Govil, S., & Leckie, J. O. (2016). Optimized Adaptive Scheduling of a Manufacturing Process System with Multi-skill Workforce and Multiple Machine Types: An Ontology-based, Multi-agent Reinforcement Learning Approach. Procedia CIRP, 57, 55–60. https://doi.org/10.1016/j.procir.2016.11.011

[71] Rinciog, A., & Meyer, A. M. (2022). Towards standardising reinforcement learning approaches for production scheduling problems. Procedia CIRP, 107, 1112–1119.

[72] Sakr, A. H., AboElHassan, A., Yacout, S., & Bassetto, S. (2021). Simulation and deep reinforcement learning for adaptive dispatching in semiconductor manufacturing systems. Journal of Intelligent Manufacturing, 34(3), 1311–1324. https://doi.org/10.1007/s10845-021-01851-7

[73] Schuh, G., Schmitz, S., Maetschke, J., Janke, J., & Eisbein, H. (n.d.). Application of a Reinforcement Learning-based Automated Order Release in Production. CONFERENCE ON PRODUCTION SYSTEMS AND LOGISTICS CPSL 2023. https://www.repo.uni-hannover.de/bitstream/handle/123456789/13610/Schuh_2023_CPSL-Application_of_a_Reinforcement_Learning-based_Automated_Order_Release.pdf?sequence=1&isAllowed=y

[74] Shi, D., Fan, W., Xiao, Y., Lin, T., & Xing, C. (2020). Intelligent scheduling of discrete automated production line via deep reinforcement learning. International Journal of Production Research, 58(11), 3362–3380. https://doi.org/10.1080/00207543.2020.1717008

[75] Su, J., Huang, J., Adams, S., Chang, Q., & Beling, P. A. (2022). Deep multi-agent reinforcement learning for multi-level preventive maintenance in manufacturing systems. Expert Systems With Applications, 192, 116323. https://doi.org/10.1016/j.eswa.2021.116323

[76] T. Zhou, D. Tang, H. Zhu and L. Wang, "Reinforcement Learning With Composite Rewards for Production Scheduling in a Smart Factory," in IEEE Access, vol. 9, pp. 752-766, 2021, doi: 10.1109/ACCESS.2020.3046784.

[77] Tang, J., & Salonitis, K. (2021). A deep reinforcement learning based scheduling policy for reconfigurable manufacturing systems. Procedia CIRP, 103, 1–7. https://doi.org/10.1016/j.procir.2021.09.089

[78] Tang, J., Haddad, Y., & Salonitis, K. (2022). Reconfigurable manufacturing system scheduling: a deep reinforcement learning approach. Procedia CIRP, 107, 1198–1203. https://doi.org/10.1016/j.procir.2022.05.131

[79] Tian, Y., & Chang, Q. (2022). User-guided motion planning with reinforcement learning for human-robot collaboration in smart manufacturing. Expert Systems With Applications, 209, 118291. https://doi.org/10.1016/j.eswa.2022.118291

[80] Viharos, Z. J., & Jakab, R. B. (2021). Reinforcement learning for statistical process control in manufacturing. Measurement, 182, 109616. https://doi.org/10.1016/j.measurement.2021.109616

[81] Waldomiro Ferreira, Cristiano Cavalcante, Phuc Do Van. Deep reinforcement learning-based maintenance decision-making for a steel production line. 31st European Safety and Reliability Conference, ESREL 2021, Sep 2021, Angers, France. ⟨10.3850/978-981-18-2016-8_600-cd⟩. ⟨hal-03379905⟩

[82] Wang, J., Gao, P., Zheng, P., Zhang, J., & Ip, W. H. (2021). A fuzzy hierarchical reinforcement learning based scheduling method for semiconductor wafer manufacturing systems. Journal of Manufacturing Systems, 61, 239–248. https://doi.org/10.1016/j.jmsy.2021.08.008

[83] Wang, X., Lin, Z., Liu, Y., & Laili, Y. (2023). An improved deep reinforcement learning-based scheduling approach for dynamic task scheduling in cloud manufacturing. International Journal of Production Research, 1–17. https://doi.org/10.1080/00207543.2023.2253326

[84] Wang, X., Zhang, L., Liu, Y., & Zhao, C. (2023). Logistics-involved task scheduling in cloud manufacturing with offline deep reinforcement learning. Journal of Industrial Information Integration, 34, 100471. https://doi.org/10.1016/j.jii.2023.100471

[85] Wang, X., Zhang, L., Liu, Y., Li, F., Chen, Z., Zhao, C., & Bai, T. (2022). Dynamic scheduling of tasks in cloud manufacturing with multi-agent reinforcement learning. Journal of Manufacturing Systems, 65, 130–145. https://doi.org/10.1016/j.jmsy.2022.08.004

[86] Wang, X., Zhang, L., Liu, Y., Zhao, C., & Wang, K. (2022). Solving task scheduling problems in cloud manufacturing via attention mechanism and deep reinforcement learning. Journal of Manufacturing Systems, 65, 452–468. https://doi.org/10.1016/j.jmsy.2022.08.013

[87] Waschneck, B., Reichstaller, A., Belzner, L., Altenmüller, T., Bauernhansl, T., Knapp, A., & Kyek, A. (2018). Optimization of global production scheduling with deep reinforcement learning. Procedia CIRP, 72, 1264–1269. https://doi.org/10.1016/j.procir.2018.03.212

[88] Wesendrup, K., & Hellingrath, B. (2023). Post-prognostics demand management, production, spare parts and maintenance planning for a single-machine system using Reinforcement Learning. Computers & Industrial Engineering, 179, 109216. https://doi.org/10.1016/j.cie.2023.109216

[89] X. -Y. Ye, Z. -W. Liu, M. Chi, M. -F. Ge and Z. Xi, "Demand Response optimization of Cement Manufacturing Industry Based on Reinforcement Learning Algorithm," 2022 IEEE International Conference on Cyborg and Bionic Systems (CBS), Wuhan, China, 2023, pp. 402-406, doi: 10.1109/CBS55922.2023.10115387.

[90] Xia, K., Sacco, C., Kirkpatrick, M., Saidy, C., Nguyen, L. M., Kircaliali, A., & Harik, R. (2021). A digital twin to train deep reinforcement learning agent for smart manufacturing plants:

Environment, interfaces and intelligence. Journal of Manufacturing Systems, 58, 210–230. https://doi.org/10.1016/j.jmsy.2020.06.012

[91] Xiong, J., Guo, P., Wang, Y., Xing, M., Zhang, J., Qian, L., & Yu, Z. (2023). Multi-agent deep reinforcement learning for task offloading in group distributed manufacturing systems. Engineering Applications of Artificial Intelligence, 118, 105710. https://doi.org/10.1016/j.engappai.2022.105710

[92] Yang, H., Li, W., & Wang, B. (2021). Joint optimization of preventive maintenance and production scheduling for multi-state production systems based on reinforcement learning. Reliability Engineering & System Safety, 214, 107713. https://doi.org/10.1016/j.ress.2021.107713

[93] Ye, Z., Cai, Z., Yang, H., Si, S., & Zhou, F. (2023). Joint optimization of maintenance and quality inspection for manufacturing networks based on deep reinforcement learning. Reliability Engineering & System Safety, 236, 109290. https://doi.org/10.1016/j.ress.2023.109290

[94] Yun, L., Wang, D., & Li, L. (2023). Explainable multi-agent deep reinforcement learning for real-time demand response towards sustainable manufacturing. Applied Energy, 347, 121324. https://doi.org/10.1016/j.apenergy.2023.121324

[95] Z. Wu, Y. Wang and L. Jia, "A Dynamic Chemical Production Scheduling Method based on Reinforcement Learning," 2022 China Automation Congress (CAC), Xiamen, China, 2022, pp. 4841-4846, doi: 10.1109/CAC57257.2022.10055985.

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

[97] Zhang, Q., Yin, G., & Wang, L. Y. (2020). Two-time scale reinforcement learning and applications to production planning. Iet Control Theory and Applications, 14(19), 3052–3061. https://doi.org/10.1049/iet-cta.2020.0049

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

[99] Zheng, P., Xia, L., Li, C., Li, X., & Liu, B. (2021). Towards Self-X cognitive manufacturing network: An industrial knowledge graph-based multi-agent reinforcement learning approach. Journal of Manufacturing Systems, 61, 16–26. https://doi.org/10.1016/j.jmsy.2021.08.002

[100] Zhou T, Zhu H, Tang D, et al. Reinforcement learning for online optimization of job-shop scheduling in a smart manufacturing factory. Advances in Mechanical Engineering. 2022;14(3). doi:10.1177/16878132221086120

[101] Zhou, L., Zhang, L., & Horn, B. K. P. (2020). Deep reinforcement learning-based dynamic scheduling in smart manufacturing. Procedia CIRP, 93, 383–388. https://doi.org/10.1016/j.procir.2020.05.163

[102] Zhu, H., Zhang, Y., Liu, C., & Shi, W. (2022). An Adaptive Reinforcement Learning-Based Scheduling Approach with Combination Rules for Mixed-Line Job Shop Production. Mathematical Problems in Engineering, 2022, 1–14. https://doi.org/10.1155/2022/1672166

[103] Zimmerling, C., Poppe, C., Stein, O., & Kärger, L. (2022). Optimisation of manufacturing process parameters for variable component geometries using reinforcement learning. Materials & Design, 214, 110423. https://doi.org/10.1016/j.matdes.2022.110423

[104] Zinn, J., Vogel-Heuser, B., & Gruber, M. (2021). Fault-Tolerant control of programmable logic Controller-Based production systems with deep reinforcement learning. Journal of Mechanical Design, 143(7). https://doi.org/10.1115/1.4050624