applied sciences

MDPI

Article

Machine Learning in Manufacturing towards Industry 4.0: From 'For Now' to 'Four-Know'

Tingting Chen [1,*], Vignesh Sampath [2], Marvin Carl May [3], Shuo Shan [1], Oliver Jonas Jorg [4], Juan José Aguilar Martín [5], Florian Stamer [3], Gualtiero Fantoni [4], Guido Tosello [1] and Matteo Calaon [1]

1 Department of Civil and Mechanical Engineering, Technical University of Denmark, 2800 Kongens Lyngby, Denmark
2 Autonomous and Intelligent Systems Unit, Tekniker, Member of Basque Research and Technology Alliance, 20600 Eibar, Spain
3 wbk Institute of Production Science, Karlsruhe Institute of Technology (KIT), Kaiserstr. 12, 76131 Karlsruhe, Germany
4 Department of Civil and Industrial Engineering, University of Pisa, 56122 Pisa, Italy
5 Department of Design and Manufacturing Engineering, School of Engineering and Architecture, University of Zarazoga, 50009 Zaragoza, Spain
* Correspondence: tchen@dtu.dk; Tel.: +45-5028-1068

Abstract: While attracting increasing research attention in science and technology, Machine Learning (ML) is playing a critical role in the digitalization of manufacturing operations towards Industry 4.0. Recently, ML has been applied in several fields of production engineering to solve a variety of tasks with different levels of complexity and performance. However, in spite of the enormous number of ML use cases, there is no guidance or standard for developing ML solutions from ideation to deployment. This paper aims to address this problem by proposing an ML application roadmap for the manufacturing industry based on the state-of-the-art published research on the topic. First, this paper presents two dimensions for formulating ML tasks, namely, 'Four-Know' (Know-what, Know-why, Know-when, Know-how) and 'Four-Level' (Product, Process, Machine, System). These are used to analyze ML development trends in manufacturing. Then, the paper provides an implementation pipeline starting from the very early stages of ML solution development and summarizes the available ML methods, including supervised learning methods, semi-supervised methods, unsupervised methods, and reinforcement methods, along with their typical applications. Finally, the paper discusses the current challenges during ML applications and provides an outline of possible directions for future developments.

Keywords: machine learning; Industry 4.0; manufacturing; artificial intelligence; smart manufacturing; digitization

check for updates

Citation: Chen, T.; Sampath, V.; May, M.C.; Shan, S.; Jorg, O.J.; Aguilar Martín, J.J.; Stamer, F.; Fantoni, G.; Tosello, G.; Calaon, M. Machine Learning in Manufacturing towards Industry 4.0: From 'For Now' to 'Four-Know'. *Appl. Sci.* **2023**, *13*, 1903. https://doi.org/10.3390/app13031903

Academic Editors: Alexandre Carvalho and Richard (Chunhui) Yang

Received: 23 November 2022
Revised: 18 January 2023
Accepted: 27 January 2023
Published: 1 February 2023

1. Introduction

Within the fourth industrial revolution, coined as 'Industry 4.0', the way products are manufactured is changing dramatically [1]. Moreover, the way humans and machines interact with one another in manufacturing has seen enormous changes [2], developing towards an 'Industry 5.0' notion [3]. The digitalization of businesses and production companies, the inter-connection of their machines through embedded system and the Internet of Things (IoT) [4], the rise of cobots [5,6], and the use of individual workstations and matrix production [7] are disrupting conventional manufacturing paradigms [1,8]. The demand for individualized and customized products is continuously increasing. Consequently, order numbers are surging while batch sizes diminish, to the extremes of fully decentralized 'batch size one' production. The demand for a high level of variability in production and manufacturing through Mass Customization is inevitable. Mass Customization in turn requires manufacturing systems which are increasingly more flexible and adaptable [7–9].

Machine Learning (ML) is one of the cornerstones for making manufacturing (more) intelligent, and thereby providing it with the needed capabilities towards greater flexibility and adaptability [10]. These advances in ML are shifting the traditional manufacturing era into the smart manufacturing era of Industry 4.0 [11]. Therefore, ML plays an increasingly important role in manufacturing domain together with digital solutions and advanced technologies, including the Industrial Internet of Things (IIoT), additive manufacturing, digital twins, advanced robotics, cloud computing, and augmented/virtual reality [11]. ML refers to a field of Artificial Intelligence (AI) that covers algorithms learning directly from their input data [12]. Despite most researchers focusing on finding a single suitable ML solution for a specific problem, efforts have already been undertaken to reveal the entire scope of ML in manufacturing. Wang et al. presented frequently-used deep learning algorithms along with an assessment of their applications towards making manufacturing "smart" in their 2018 survey [13]. In particular, they discussed four learning models: Convolutional Neural Networks, Restricted Boltzmann Machines, Auto-Encoders, and Recurrent Neural Networks. In their recent literature review on "Machine Learning for Industrial Applications", Bertolini et al. [12] identified, classified, and analyzed 147 papers published during a twenty-year time span from Jan. 2000 to Jan. 2020. In addition, they provided a classification on the basis of application domains in terms of both industrial areas and processes, as well as their respective subareas. Within these domains, the authors analyzed the different trends concerning supervised, unsupervised, and reinforced learning techniques, including the most commonly used algorithms, Neural Networks (NNs), Support Vector Machine (SVM), and Tree-Based (TB) techniques. The goal of another literature review from Dogan and Birant [14] was to provide a sound comprehension of the major approaches and algorithms from the fields of ML and data mining (DM) that have been used to improve manufacturing in the recent past. Similarly, they investigated research articles from the period of the past two decades and grouped the identified articles under four main subjects: scheduling, monitoring, quality, and failure.

While these classifications and trend analyses provide an excellent overview of the extent of ML applications in manufacturing, they mainly focus on introducing ML algorithms; the implementation of ML solution for different tasks in an industrial environment from scratch has not yet been fully discussed. In general, a comprehensive formulation of industrial problems prior to the development of ML solutions seems lacking. Therefore, the issue we aim to address in this paper is how ML can be implemented to improve manufacturing in the transition towards Industry 4.0. From this issue, we derive the following research questions:

- RQ1: How does ML benefit manufacturing, and what are the typical ML application cases?
- RQ2: How are ML-based solutions developed for problems in manufacturing engineering?
- RQ3: What are the challenges and opportunities in applying ML in manufacturing contexts?

To answer these research questions, more than a thousand research articles retrieved from two well-known research databases were systematically identified, screened, and analyzed. Subsequently, the articles were classified within a two-dimensional framework, which takes value-based development stages into account on one axis and manufacturing levels on the other. The development stage concerns visibility, transparency, predictive capacity, and adaptability, whereas the four manufacturing levels are product, process, machine, and system.

The rest of this paper is structured as follows. Section 1 introduces the key concepts, research questions, and motivations. Section 2 proposes the methodology of 'Four-know' and 'Four-level' to establish a two-dimensional framework for helping to formulate industrial problems effectively. Based on the proposed framework, a systematic literature review is carried out and the identified articles are analysed and classified. Section 3 describes a six-step pipeline for the application of ML in manufacturing. Section 4 explains different ML methods, presenting where and how they have been applied in manufacturing according to the prior identified research articles. Section 5 formulates common challenges and potential future directions; finally, the paper concludes in Section 6 with a summary and discussion of the authors' findings.

2. Overview of Machine Learning in Manufacturing

Despite numerous ML studies and their promising performance, it remains very difficult for non-experts working in the manufacturing industry to begin developing ML solutions for their specific problems. The first challenging part of application is to formulate the actual problems to be solved [15]. Therefore, this section aims to overcome this problem by introducing the categories of Four-Know and Four-Level to help formulate ML tasks in manufacturing and describing the benefits of applying ML in manufacturing from ML use cases categorized using the Four-Know and Four-Level concepts (RQ1). Lastly, an overview and developing trends in recent ML studies are provided as formulated by Four-Know and Four-Level.

2.1. Introduction of Four-Know and Four-Level

According to the *Acatech* Industrie 4.0 Maturity levels [16], the development towards Industry 4.0 in manufacturing can be structured into the following six successive stages: computerization, connectivity, visibility, transparency, predictive capacity, and adaptability. The first two stages, computerization and connectivity, provide the basis for digitization, while the rest are analytic capabilities required for achieving Industry 4.0. ML, as powerful data analytics tools are normally applied in the last four stages. Inspired by the *Acatech* Industrie 4.0 Maturity levels, ML studies in manufacturing can be categorized into four subjects: Know-what, Know-why, Know-when, and Know-how, which to a degree overlap with visibility, transparency, predictive capacity, and adaptability, respectively. The Four-Know definitions are presented below:

- *Know-what* deals with understanding of the current states of machines, processes, or production systems, which can help in rapid decision-making. It should be noted that Know-what goes beyond visualization of real-time data. Instead, data should be processed, analyzed, and distilled into information which enables decision-making. For instance, typical examples of Know-what in manufacturing are defect detection in quality control [17,18], fault detection in process/machine monitoring [19,20], and soft sensor modelling [21,22].
- *Know-why*, based on the information from Know-what, aims to identify inner patterns from historical data, thereby discovering the reasons for a thing happening. Know-why includes the identification of interactions among different variables [23] and the discovery of cause-effect relationship between an event and other variables [24,25]. On one hand, Know-why can indicate most important factors for understanding Know-what. On the other hand, Know-why is the prerequisite for Know-when, as the reliability of predictions is heavily dependent upon the quality of casual inference.
- *Know-when*, built on Know-why, involves timely predictions of events or prediction of key variables based on historical data, allowing the decision-maker can take actions at early stages. For instance, Know-when in manufacturing includes quality prediction based on relevant variables [26,27], predictive maintenance via detection of incipient anomalies before break-down [28,29], and predicting Remaining Useful Life (RUL) [30,31].
- *Know-how*, on the foundation of Know-when, can recommend decisions that help adapt to expected disturbance and can aid in self-optimization. Examples in manufacturing include prediction-based process control [27,32], scheduling of predictive maintenance tasks [33,34], dynamic scheduling in the flexible production [35,36], and inventory control [34].

The aim of applying ML in manufacturing is to achieve production optimization across four different levels: product, process, machine, and system. Therefore, the use cases for applying ML can be further categorized by these different levels, as shown in Figure 1 and Table 1, which answer RQ1 in terms of ML typical use cases.

Table 1. Typical ML use cases categorized by Four-Level and Four-Know.

Level	Know-What	Know-Why	Know-When	Know-How
Product	Defect detection [37], Product design [38]	Correlation between process and quality [23]	Quality prediction [26]	Quality improvement [39]
Process	Process monitoring [40]	Root cause analysis of process failure [41], Process modelling [42]	Process fault prediction [43], Process characteristics prediction [44]	Self-optimizing process planning [45], Adaptive process control [46]
Machine	Machine tool monitoring [47]	Fault diagnosis [48], Downtime prediction [49]	RUL prediction [50], Tool wear prediction [51]	Adaptive compensation of errors [52,53],
System	Anomaly detection [54]	Root cause analysis of production disturbances or casual-relationship discovery [55]	Production performance prediction [56], Human behavior control [57]	Predictive scheduling [58], Adaptive production control [59]

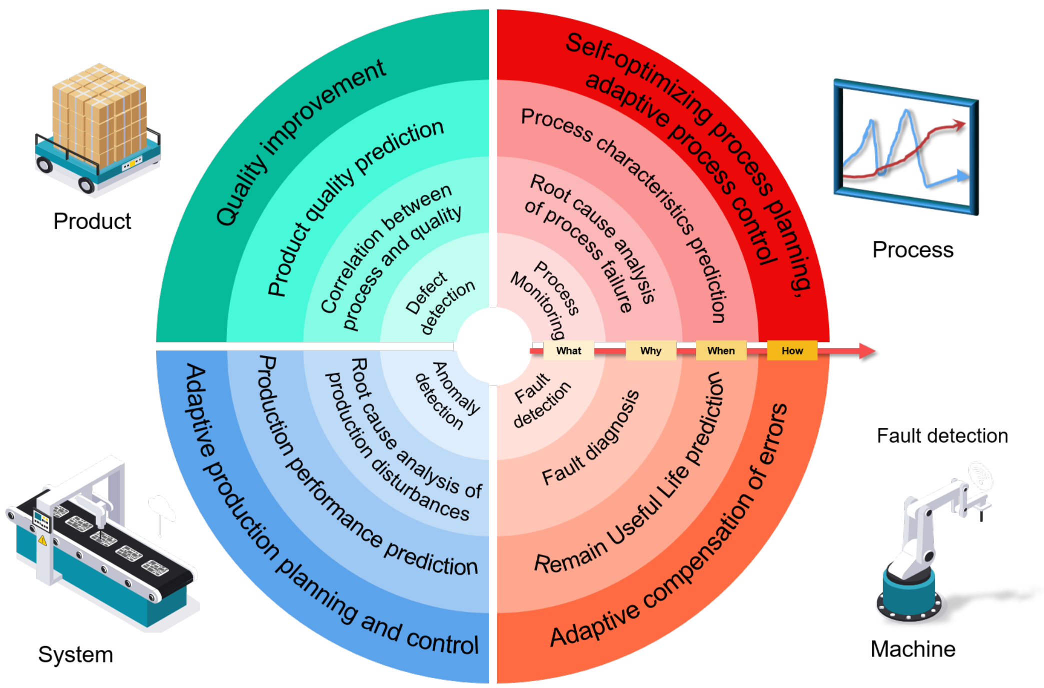

Figure 1. Four-Level and Four-Know categorization of ML applications. The Four-Know categories, from Know-what to Know-how, are respectively demonstrated by the four concentric circles, from the inner circle to the outer circle, with each circle divided into four quarters according to the Four Levels.

2.2. Literature Review Methodology

In order to address the research questions laid out in Section 1, a systematic literature review following the PRISMA methodology [60] was carried out. Two well-known research databases, Scopus (Elsevier) and Web of Science (WoS), were chosen for retrieving documents. The overall literature review process is shown in Figure 2.

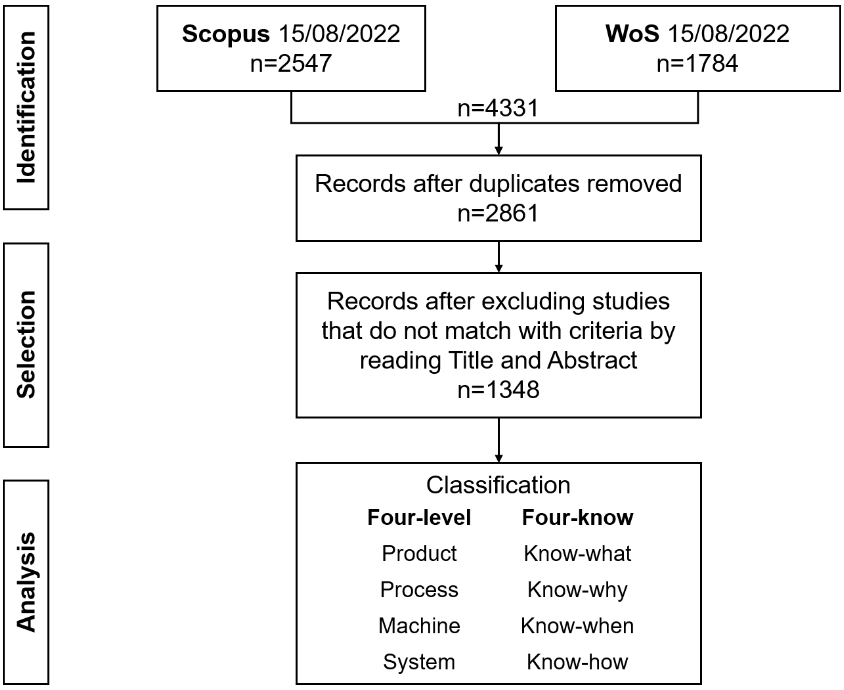

Figure 2. The overall literature review process following PRISMA. All identified documents were screened and assessed for eligibility, then subjected to Four-Level and Four-Know classification.

Table 2 shows the limitations used when performing the document search. It should be noted that the query strings were used for Title, Abstract, and Keywords as well as Keyword Plus (only in WoS).

Table 2. Limitations for document searching.

Item	Description
Query string	("manufacturing" OR "industry 4.0" OR "industrie 4.0") AND ("machine learning" OR "deep learning" OR "supervised learning" OR "semi-supervised learning" OR "unsupervised learning" OR "reinforcement learning")
Year	Published from 2018 to 2022
Language	English
Subject/Research area	Engineering
Document type	Article

Following the document search, 2547 documents were found from Scopus and 1784 from WoS. The identified publications from the two databases were merged and duplicates were removed, resulting in 2861 publications. The documents were then evaluated and selected by reading the Title and Abstract field, and articles that did not meet the following selection criteria were excluded:

- The study dealt with the context of manufacturing;
- The study dealt with ML applications in specific fields.

Therefore, conceptual models, frameworks, and studies that only focused on algorithm development were considered to be out of scope.

Finally, the remaining 1348 documents were analyzed and classified based on the Four-Level and Four-Know categories. Figure 3 shows the trend of ML applications in manufacturing over the past five years from the Four-Level perspective. Figure 4 reveals the detailed distribution of ML applications in Four-Know terms. It should be noted that because the literature review was conducted in August 2022, the actual numbers for the full year 2022 should be higher. As can be seen, there has been a gradual increase in the number of ML publications in manufacturing in all levels over the past five years. Typically, what stands out in this figure is the dominance of the product level. From Figure 4, it can be seen that recent ML applications in product level are mainly focused on Know-what and Know-when. A similar pattern can be found at the machine level. Interestingly, a considerable growth in Know-how is observed at the process and system levels compared to the others. The reason for this may be correlated with higher demand for adaptability with respect to changes on the process and system levels.

The identified documents were analyzed and classified according to their applied ML methods, providing examples for non-experts when dealing with similar tasks.

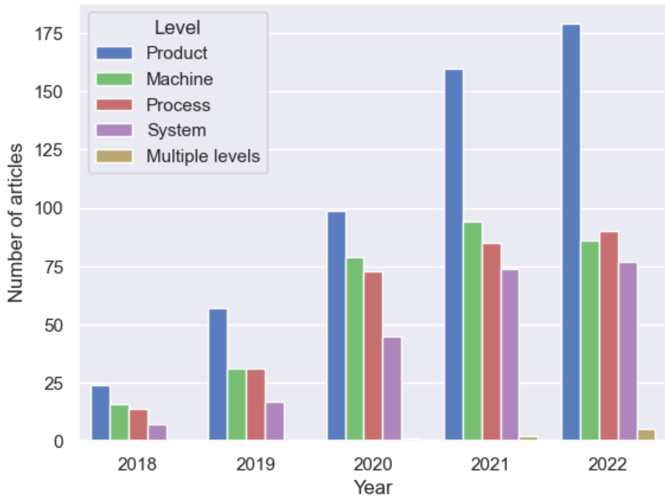

Figure 3. Trends in ML publications in manufacturing in the past five years by Four-Level grouping.

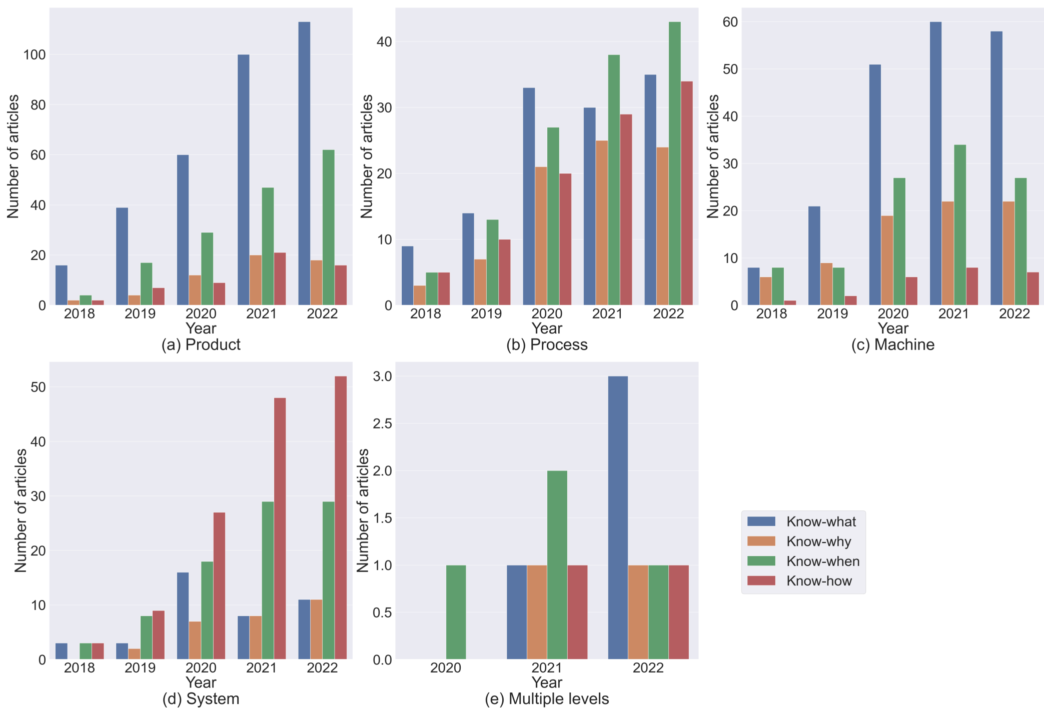

Figure 4. Four-Know development trends for each level over the past five years.

3. Pipeline of Applying Machine Learning in Manufacturing

ML is a technique capable of extracting knowledge from data automatically [12]. Increasing research on ML has shown that it is an appealing solution when tackling complex challenges. In recent years, more and more manufacturing industries have begun to leverage the benefits of ML by developing ML solutions in several industrial fields. However, despite plenty of off-the-shelf ML models, there are challenges when applying ML to real-world problems [61]. In particular, it is harder for small and medium-sized enterprises to develop in-house ML solutions, as commercial ML solutions are normally confidential and inaccessible. Therefore, this section aims to provide a pipeline for applying ML for those who are starting from scratch (RQ2). Applying machine learning in manufacturing normally involves the following six steps: (i) data collection, (ii) data cleaning, (iii) data transformation, (iv) model training, (v) model analysis, and (vi) model push, as shown in Figure 5.

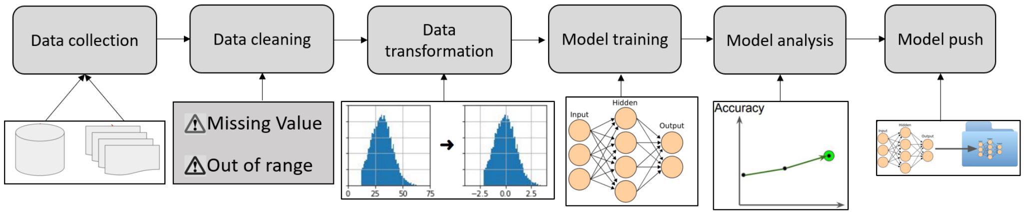

Figure 5. Pipeline of applying machine learning in manufacturing.

3.1. Data Collection

The lifeblood of any machine learning model is data. In order for an ML model to learn, clean data samples must be continuously fed into system throughout the training process. When the collected data are highly imbalanced or otherwise inadequate, the desired task may not be achievable. Data can be collected from different sources, including machines, processes, or production with the aid of sensors or external databases. In terms of data types, the data used in machine learning can be generally categorized as follows:

- *Image data*, matrices of pixels with two or more dimensions, such as gray-scale images or colored images. Image data can acquired by with vision systems, through data transformations such as simple concatenation of several one-dimensional vectors with same length, or by the transformation of images from the spatial domain to the frequency domain.
- *Tabular data* organized in a table, where normally one axis represents attributes and another axis represents observations. Tabular data are typically observed in production data, where the attributes of events of interest are collected. Though tabular data share a similar data structure with image data, the latter are more focused on one-dimensional interaction among attributes, while image data typically stress spatial interactions in both dimensions.
- *Time series data*, sequences of one or more attributes over time, with the former corresponding to univariate time series and the latter multivariate time series. In manufacturing, time series data are normally acquired with sensors whenever there is a need for monitoring time flow changes of data.
- *Text data*, including written documents with words, sentences or paragraphs. Examples of text data in manufacturing include maintenance reports on machines and descriptions of unexpected disturbances or events in production.

3.2. Data Cleaning

Real-world industrial data are highly susceptible to noisy, missing, and inconsistent data due to several factors. Low-quality noisy data can lead to less accurate ML models. Data cleaning [62] is a crucial step when organizing data into a consistent data structure across packages, and can improve the quality of the data, leading to more accurate ML models. It is usually performed as an iterative approach. Methods include filling in missing values, smoothing noisy data, removing outliers, resolving data inconsistencies, etc.

3.3. Data Transformation

Data transformation is the process of transforming unstructured raw data into data better suited for model construction. Data transformation can be broadly classified into mandatory transformations and optional quality transformations. Mandatory transformations must be carried out to convert the data into a usable format and then deliver the transformed data to the destination system. These include transforming non-numerical data into numerical data, resizing data to a fixed size, etc. It should be noted that data transformations are not always straightforward. Indeed, in certain situations data types can be interconvertible by leveraging specific processing techniques, as shown in Figure 6. For instance, univariate time series can be converted into image data using the Gramian Angular Field (GAF) or Markov Transition Field (MTF) [63] methods. Unstructured text data can be converted into tabular data via word embedding [64]. Tabular data can be transformed into image data by projecting data into a 2D space and assigning pixels, as in Deepinsight [65] or Image Generator for Tabular Data (IGTD) [66]. Image data are preferable for data analysis, as they allow the power of Convolutional Neural Networks (CNNs) [67] to be exploited.

In real-world applications, data are normally high-dimensional and redundant. When performing data modelling directly in the original high-dimensional space, the computational efficiency can be very low. Hence, it is necessary to reduce the dimensionality in order to obtain better representation for data modelling. This is achieved by feature selection, which selects the most informative feature subset from raw data, or feature extraction, which generates new lower-dimensional features. After feature engineering, features are either manually designed, so-called "handcrafted features" [68], or automatically learned from data, so-called "automatic features". Handcrafted features are heavily dependent on domain knowledge, and normally have physical meaning. However, these features are highly subjective [69] and inevitably lack implicit key features [70,71].

By contrast, automatic features driven by data require no prior knowledge. Therefore, they have been gaining increasing research attention in recent years. Conventionally, automatic features are obtained by linear transformations such as Principle Component Analysis (PCA) [72] or Independent Component Analysis (ICA) [73]. However, with the development of Artificial Neural Networks (ANNs), direct learning of implicit features has become possible by optimizing the loss function. Thus, neural networks have gradually developed into an end-to-end solution where knowledge is directly learned from raw data without human effort. Typically, CNNs [74] and Recurrent Neural networks (RNNs) [75] are used for image data and time series data, respectively.

A summary of typical features for different data types can be seen in Table 3.

Table 3. Typical features for different data types.

Data Type	Handcrafted Features	Automatic Features
Image data	LBP [76], SIFT [77], HOG [78]	ICA, CNNs
Tabular data	feature selection	PCA, ICA, ANNs
Time series data	Time domain: mean, min, max, etc. Frequency domain: power spectrum [78] Time-frequency domain: DWT [79], STFT [80]	ICA, RNNs
Text data	Bag of Words (BoW) [81]	Word2vec [82]

3.4. Model Training

After selecting the features, it is necessary to form the correct data structure for each individual ML model used in the subsequent steps. Note that different ML algorithms might require different data models for the same task. Furthermore, results can be improved through normalization or standardization. Then, the ML models can be applied in the actual modelling phase. The first step in training a machine learning model typically involves selecting a model type that is appropriate for the nature of the data and the problem at hand. After a model has been chosen, it can be trained by providing it with the training data and using an optimization algorithm to find the set of parameters that provide the best performance on those data. Depending on the task, either unsupervised, semi-supervised, supervised, or reinforcement learning can be applied. These are individually introduced in the following section.

3.5. Model Analysis

Analysis of model performance is an important step in choosing the right model. This stage emphasizes how effective the selected model will perform in the future and helps to make the final decision with regard to model selection. Performance analysis evaluates models using different metrics, e.g., accuracy, precision, recall, and F1-score (the weighted average of precision and recall) for classification tasks and the root mean square error (RMSE) for regression tasks.

3.6. Model Push

Although state-of-the-art ML models improve predictive performance, they contain millions of parameters, and consequently require a large number of operations per inference. Such computationally intensive models make deployment in low-power or resource-constrained devices with strict latency requirements quite difficult. Several methods, including model pruning [83], model quantization [84], and knowledge distillation [85], have been suggested in the literature as ways to compress these dense models.

Overall, In the context of manufacturing applications, data collection, data cleaning, data transformation, model training, model analysis, and model push are key steps in the implementation of utilizing historical data with ML in order to optimize production and improve efficiency, quality, and productivity. For instance, data collection involves gathering data from various sources, such as sensor data, production logs, and quality control records. Data cleaning involves removing any errors, inconsistencies, or irrelevant information from the data. Data transformation involves preparing the data for analysis via formatting in a way that is suitable for the chosen model. Model training involves using the cleaned and transformed data to train a machine learning model. Model analysis involves evaluating the performance of the model and identifying any areas for improvement. Model push involves deploying the model in a production environment and making predictions or decisions based on the model. All of these steps are critical to ensuring that the results from ML models are accurate, reliable, and useful for manufacturing production.

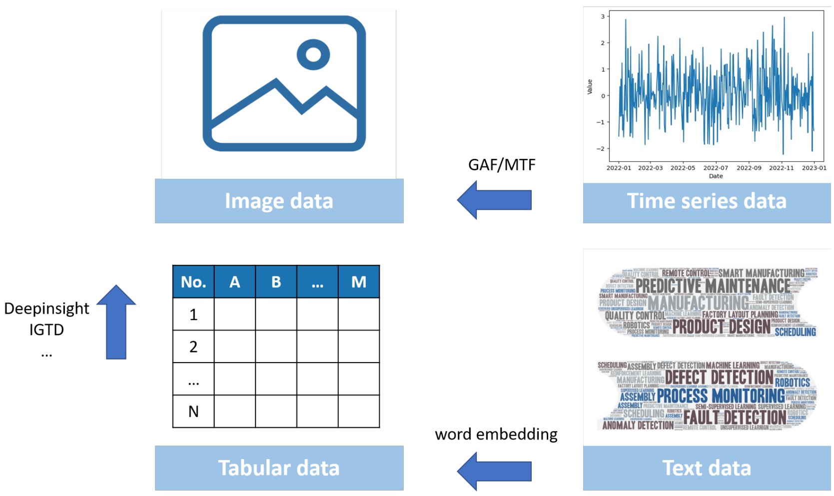

Figure 6. Data types used in ML and their convertibility.

4. Machine Learning Methods and Applications

Model development is the core of ML-based solutions, as the selection of an ML model plays a critical roles in the outcome. Therefore, this section aims to provide a comprehensive overview of ML methods and their potential possibilities in manufacturing applications, including supervised learning methods, semi-supervised learning methods, unsupervised learning methods, and reinforcement learning methods. In addition, example typical applications for each category of ML method are listed to support model selection.

4.1. Supervised Learning Methods

Supervised learning methods aim to learn an approximation function f that can map inputs x to outputs y with the guidance of annotations $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$. In supervised learning, the algorithm analyzes a labeled dataset and derives an inferred function which can be applied to unseen samples. It should be noted that labeled dataset is a necessity for supervised learning, and as such it requires a large amount of data and high labeling costs. Supervised learning methods are generally used for dealing with two problems, namely, regression and classification. The difference between regression and classification is in the data type of the output variables; 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.

Tree-based methods: Tree-based methods aim at partitioning the feature space into several regions until the datapoints in each region share a similar class or value, as depicted in Figure 7. After space partitioning, a series of if–then rules with a tree-like structure can be obtained and used to determine the target class or value. Compared with the black-box models in other supervised methods, Tree-based methods are easily understandable models that offer better model interpretability. Decision trees [86], in which only a single tree is established, are the most basic of tree-based methods. It is simple and effective to train a decision tree, and the results are intuitively understandable, though this approach is very prone to overfitting. A tree ensemble is an extension of the decision tree concept. Instead of establishing a single tree, multiple trees are established in parallel or in sequence, referred to as bagging [87] and boosting [88], respectively. Commonly used tree ensemble methods include Random Forest [89], Adaptive Boosting (AdaBoost) [88], and Extreme Gradient Boosting (XGBoost) [90].

Thanks to their better model interpretability, tree-based methods can be used to identify the most important factors leading up to events. Their possible applications in manufacturing are mainly in the Know-why and Know-when stages. For instance, examples of Know-why tasks with tree-based methods at the product and machine level include identifying the influencing factors that lead to quality defects [91] or machine failure [92], thereby allowing the manufacturer to diagnose problems effectively. In addition, the identified important factors when using tree-based methods can help in further predicting target values such as product quality [93](Know-when, product level) or events of interest before they happen, such as machine breakdown [31] (Know-when, machine level).

Probabilistic-based methods: For a given input, probabilistic-based methods provide probabilities for each class as the output. Probabilistic models are able to explain the uncertainties inherent to data, and can hierarchically build complex models. Widely used probabilistic-based methods include Bayesian Optimization (BO) [94] and Hidden Markov Models (HMM) [95].

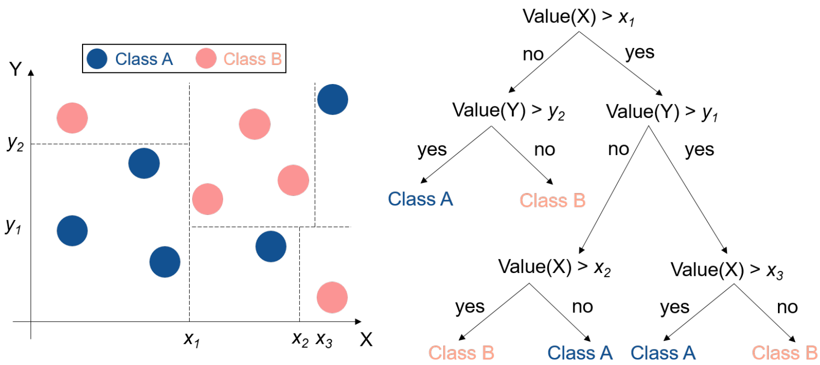

Figure 7. The principle of a decision tree. As shown, the feature space is partitioned into several rectangles in which the input point can find the corresponding class.

The dependencies among different variables can be well captured by Bayesian networks [94], enabling a greater likelihood of predicting the target. This can be potentially beneficial for manufacturing when it comes to Know-what and Know-when tasks, for instance, detection or prediction of events such as quality issues [96] (product level), machine failure [97] (machine level), or dynamic process modelling [98] (process level).

Markov chains [95], on the other hand, are a type of probabilistic model that describe a sequence of possible events in which the probability of each event depends only on the state attained in the previous event. Markov chains can be utilized in manufacturing to model and analyze the behavior of systems (Know-why, system level) such as production lines [99] or supply chains [100]. In addition, the capability of predicting future states with Markov chains enables applications predicting joint maintenance in production systems [101] (Know-when, system level) and optimizing production scheduling [102] (Know-how, system level).

Kernel-based methods: As depicted in Figure 8, kernel-based methods utilize a defined kernel function to map input data into a high-dimensional implicit feature space [103]. Instead of computing the targeted coordinates, kernel-based methods normally compute the inner product between a pair of data points in the feature space. However, kernel-based methods have low efficiency, especially with respect to large-scale input data. Due to the promising capability of kernel-based methods in classification and regression, they can be utilized in the Know-what and Know-when stages in manufacturing, such as defect detection [104] (Know-what, product level), quality prediction [105] (Know-when, product level), and wear prediction in machinery [106] (Know-when, machine level). There are different types of kernel-based methods in supervised learning, such as SVM [107] and Kernel–Fisher discriminant analysis (KFD) [108].

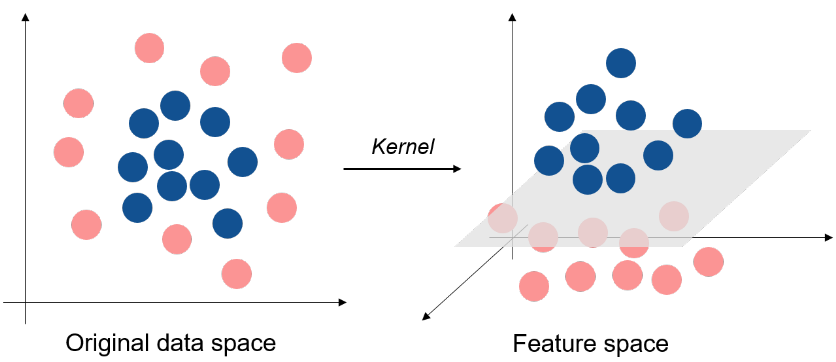

Figure 8. The principle of kernel-based methods. Using a kernel, the linearly inseparable input data are transformed to another feature space in which they become linearly separable.

Neural-network-based methods: Inspired by biological neurons and their ability to communicate with other connected cells, neural network-based methods employ artificial neurons. A typical neural network, such as ANNs, consists of an input layer, hidden layer, and output layer, as illustrated in Figure 9. Common ANNs types include CNNs [109], RNNs [110], and Deep Belief Network (DBN) [111].

Thanks to their powerful feature extraction capability when using matrix-like data, CNNs are widely used for image processing. In terms of possible applications in manufacturing, CNNs can be used in the Know-what stage to perform image-based quality control [112] (Know-what, product level) or image-based process monitoring [113] (Know-what, process level). In addition, by converting time series data from sensors to 2D images [114], CNNs can be used to detect and diagnosis machine failure as well.

RNNs are typically used to process sequential input data such as time series data or sequential images. Therefore, in terms of possible applications in manufacturing, RNNs are well-suited to the Know-when stage for analyzing sensor data or live images from machines, processes, or production systems. For instance, RNNs can enable the real-time performance prediction, such as the remaining useful life of machinery [115] (Know-when, machine level), process behavior prediction [116] (Know-when, process level), or the prediction of production indicators for real-time production scheduling [117] (Know-when, system level).

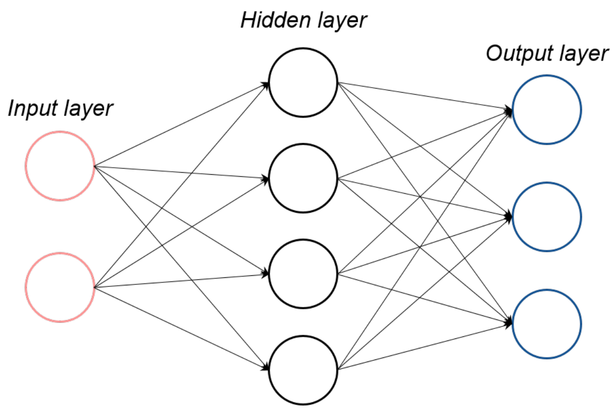

Figure 9. The scheme of an ANN, which normally consists of an input layer, hidden layer and output layer.

The typical supervised learning approaches applied in manufacturing are summarized in Table A1.

4.2. Unsupervised Learning Methods

Unsupervised learning algorithms aim to identify patterns in data sets containing data points that are not labeled. Unsupervised learning eliminates the need for labeled data and manual feature engineering, allowing for more general, flexible, and automated ML methods. As a result, unsupervised learning methods draw patterns and highlight areas of interest, revealing critical insight into the production process and opportunities for improvement. This can allow manufacturers to make better production-focused decisions, driving their business forward. The primary goal of unsupervised learning is to identify hidden and interesting patterns in unlabeled data. In terms of principles, there are three types of unsupervised tasks: Dimension Reduction [118,119], Clustering [120], and Association Rules [121]. Many aspects of unsupervised learning can be beneficial in manufacturing applications. First, clustering algorithms can be used to identify outliers in manufacturing data. Another aspect is to handle high dimensional data, e.g., for manufacturing cost estimation, quality improvement methodologies, production process optimization, better understanding of the customer's data, etc. Usually, a dimensional reduction support algorithm is required to handle data complexity and high dimensionality. Finally, it is challenging to perform root cause analysis in large-scale process execution due to the complexity of services in data centers. Association rule-based learning can be employed to conduct root cause analysis and to identify correlations between variables in a dataset.

Dimensional reduction is the process of converting data from a high-dimensional space to a low-dimensional space while preserving important characteristics of the original data.

Principal component analysis (PCA) [118]: The main idea of PCA is to minimize the number of interrelated variables in a dataset while preserving as much of the dataset's inherent variance as possible. A new set of variables, called principal components (PCs), are generated; these are uncorrelated and sorted such that the first few variables retain the majority of the variance included in all of the original variables. A pictorial representation of PCA is shown in Figure 10.

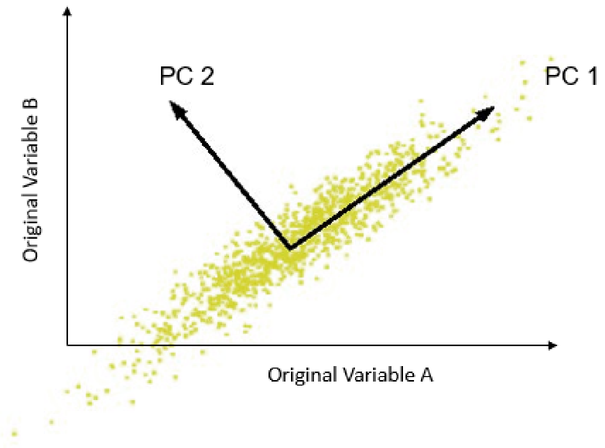

Figure 10. Principal Component Analysis.

The five steps below can be used to condense the entire process of extracting principal components from a raw dataset.

1. Say we wish to condense d features in our data matrix X to k features. The first step is to standardize the input data:
$$z = x - \mu/\sigma$$
where μ is the mean and σ is the standard deviation.
2. Next, it is necessary to find the covariance matrix of the standardized input data. The covariance of variables X and Y can be written as follows:
$$\mathrm{cov}(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}(Xi - \vec{x})(Yi - \vec{y}). \tag{1}$$
3. The third steps is to find all of the eigenvalues and eigenvectors of the covariance matrix:
$$A\vec{v} = \lambda\vec{v} \tag{2}$$
$$A\vec{v} - \lambda\vec{v} = 0 \tag{3}$$
$$\vec{v}(A - \lambda I) = 0. \tag{4}$$
4. Then, the eigenvector corresponding to the largest eigenvalue is the direction with the maximum variance, the eigenvector corresponding to second-largest eigenvalue is the direction with the second maximum variance, etc.
5. To obtain k features, it is necessary to multiply the original data matrix by the matrix of eigenvectors corresponding to the k largest eigenvalues.

PCA is particularly useful for processing manufacturing data, which typically have a large number of variables, making it difficult to identify patterns and trends. A variety of applications of PCA in manufacturing are listed below:

1. Quality improvement (Know-why, product level): by analyzing the variations of a product's features, PCA can be used to identify the causes of product defects [122].
2. Machine monitoring (Know-why, machine level): by analyzing sensor data from a machine, PCA can be used to detect incipient patterns in the data that indicate potential issues with the machinery, such as wear and tear [123].
3. Process optimization (Know-why, process level): by analyzing variations in the process data, PCA can be used to identify the most important factors that affect the process, allowing the manufacturer to optimize the process and thereby reduce costs [124].

Autoencoder (AE) [119] is another popular method for reducing the dimensionality of high-dimensional data. AE alone does not perform classification; instead, it provides a compressed feature representation of high-dimensional data. The typical structure of AE consists of an input layer, one hidden or encoding layer, one reconstruction or decoding layer, and an output layer. The training strategy of AE includes encoding input data into a latent representation that can reconstruct the input. To learn a compressed feature representation of input data, AE tries to reduce the reconstruction error, that is, to minimize the difference between the input and output data. An illustration of AE is shown in Figure 11.

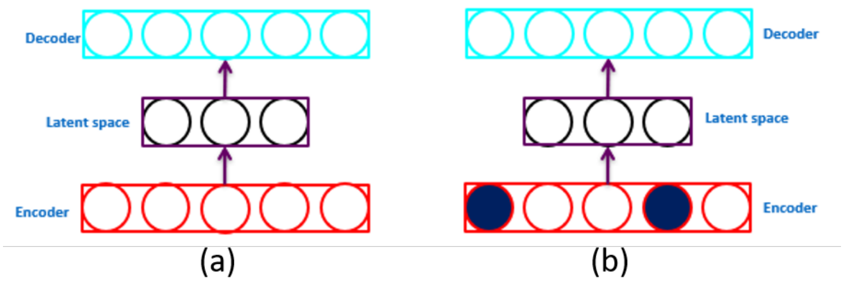

Figure 11. A pictorial representation of (**a**) an Autoencoder and (**b**) a Denoising Autoencoder. An autoencoder is trained to reconstruct its input, while a denoising autoencoder is trained to reconstruct a "clean" version of its input from a corrupted or "noisy" version of the input.

There are different types autoencoders that can be used for high-dimensional data. *Stacked Autoencoder* (SAE) [119] is built by stacking multiple layers of AEs in such a way that the output of one layer serves as the input of the subsequent layer. *Denoising autoencoder* (DAE) [125] is a variant of AE that has a similar structure except for the input data. In DAE, the input is corrupted by adding noise to it; however, the output is the original input signal without noise. Therefore, unlike AE, DAE has the ability to recover the original input from a noisy input signal. *Convolutional autoencoder* [126] is another interesting variant of AE, employing convolutional layers to encode and decode high-dimensional data.

AEs can be used for a variety of applications in manufacturing, such as:

1. Anomaly detection (Know-what): an AE can be trained to reconstruct normal data and detect abnormal data by measuring the reconstruction error, which allows the manufacturer to detect and address issues such as product defects [124] and machinery failure [127].
2. Feature selection (Know-why): an AE can be used to identify the most important features in the data and remove the noise and irrelevant information, which can be used for diagnosis of product defects or to detect events of interests [128].
3. Dimensionality reduction: an AE can be used to reduce the dimensionality of large and complex datasets, making it easier to identify patterns and trends [129].

Furthermore, AEs can be used in conjunction with other techniques, such as clustering or classification, to improve the accuracy of prediction and enhance the interpretability of the results [130]. Additionally, AEs can be used for data visualization. By reducing the dimensionality of the data, AEs allow high-dimensional data to be visualized clearly and interpretably [129] in a way that can be easily understood by non-technical stakeholders.

Clustering: The objective of clustering is to divide the set of datapoints into a number of groups, ensuring that the datapoints within each group are similar to one another and different from the datapoints in the other groups. Clustering methods are powerful tools, allowing manufacturers examine large and complex datasets and gain meaningful insights. There are different clustering methods available, each with their own strengths and weaknesses, and the choice of method depends on the characteristics of the data and the problem to be solved. Among the widely used clustering methods are *Centroid-based Clustering* [120], *Density-based Clustering* [131], *Distribution-based Clustering* [132], and *Hierarchical Clustering* [133]. Clustering algorithms have a wide range of applications in manufacturing. For instance, clustering can be used to group manufactured inventory parts according to different features [134] (Know-what). The obtained clusters can be used as a guideline for warehouse space optimization [135]. Clustering can be used for anomaly detection [136] (Know-what) and process optimization [137] (Know-how), and can be used in conjunction with other techniques to improve the interpretability of results.

Association rule-based learning [121]: Association rule-based learning is an unsupervised data-mining technique that finds important interactions among variables in a dataset. It is capable of identifying hidden correlations in datasets by measuring degrees of similarity. Hence, association rule-based learning is suitable in the Know-why stage in manufacturing. For instance, association rule-based learning can be utilized to accurately depict the relationship between quantifiable shop floor indicators and appropriate causes of action under various conditions of machine utilization (Know-why, system level), which can be used to establish an appropriate management strategy [138].

4.3. Semi-Supervised Learning Methods

Unsupervised learning methods do not have any input guidance during training, which reduces labeling costs; however, their performance is normally less accurate. Therefore, semi-supervised learning methods can be used to take advantage of the accuracy achieved by supervised learning while limiting costs thanks to the reduction in labeling effort. Therefore, researchers have turned to data augmentation [139,140] to enlarge dataset, with the inputs and labels generated massively based on the existing dataset in a controlled way while incurring no extra cost in the labeling phase. Taking an image with its label as an example, it can be enriched by basic transformations such as rotation, translation, flipping, noise injection, etc. It can be enriched by adversarial data augmentation, such as by generating synthetic dataset using generative models, e.g., Generative Adversarial Network (GAN) [141] and Variational AutoEncoder (VAE) [142], thereby obtaining new images for training ML models at low cost. However, the improvements obtainable with data augmentation are limited, and more real data are better than more synthetic data [143]. Therefore, increasing attention is being paid to the combination of supervised learning and unsupervised learning, namely, semi-supervised learning, in which both unlabeled data and labeled data are leveraged during training.

Semi-supervised learning methods can be generally divided into two groups: data augmentation-based methods and semi-supervised mechanism-based methods. An overview of semi-supervised methods is provided in Figure 12.

Data augmentation: through data augmentation, labeled data can be enlarged and augmented by adding model predictions of newly unlabeled data with high confidence as pseudo-labels, as shown in Figure 13. However, the model continues to be run in a fully supervised manner. In addition, the quality of the pseudo-labels can highly affect model performance, and incorrect pseudo-labels with high confidence are inevitable due to their nature. To improve the quality of pseudo-labels, there are hybrid methods combining pseudo-labels and consistency regularization, such as MixMatch [144] and FixMatch [145]. Nevertheless, data augmentation-based methods are simple, and there is no need to carefully design the loss. Therefore, data augmentation-based methods can be potentially useful for non-experts in manufacturing for enlarging labeled dataset when it is easy to collect massive amounts of unlabeled data.

Semi-supervised mechanisms: by contrast, semi-supervised mechanism-based methods are more focused on the mechanism of utilizing both labeled data and unlabeled data. The principle of semi-supervised mechanisms is illustrated in Figure 14, where both labeled data and unlabeled data can be model inputs while their losses are calculated in a different way. Semi-supervised mechanism-based methods can be further categorized into consistency-based methods, graph-based methods, and generative-based methods.

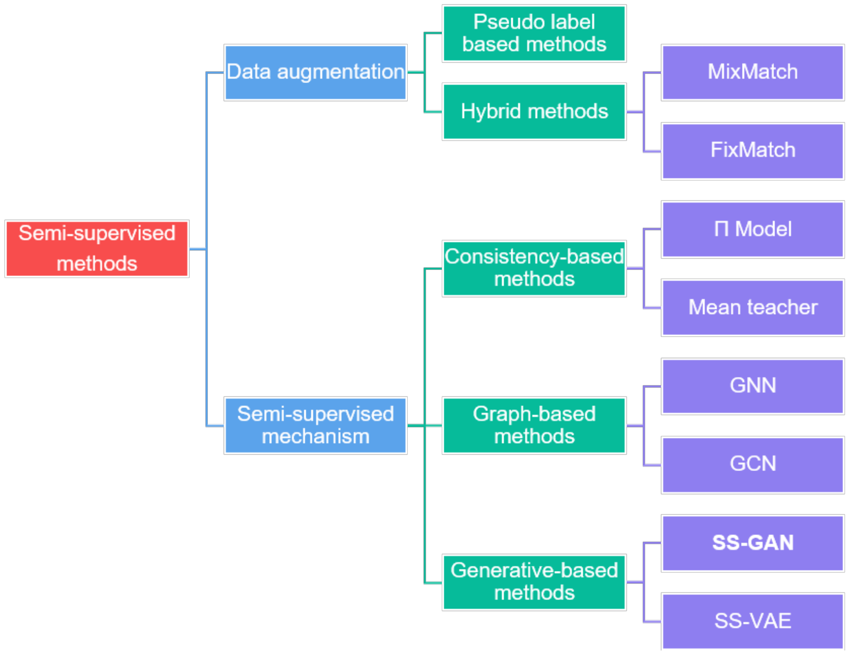

Figure 12. Overview of semi-supervised methods.

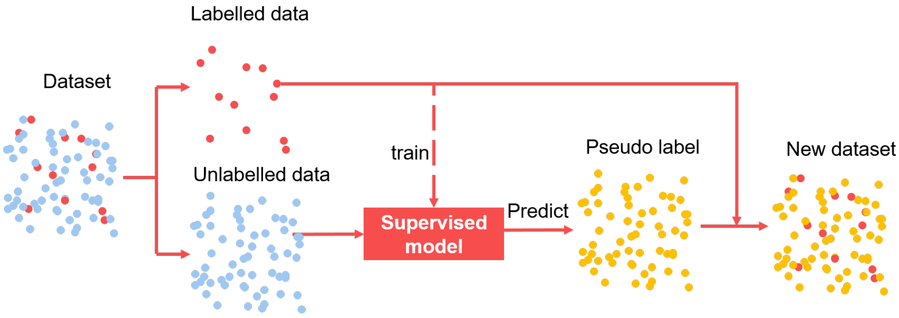

Figure 13. Data augmentation-based methods.

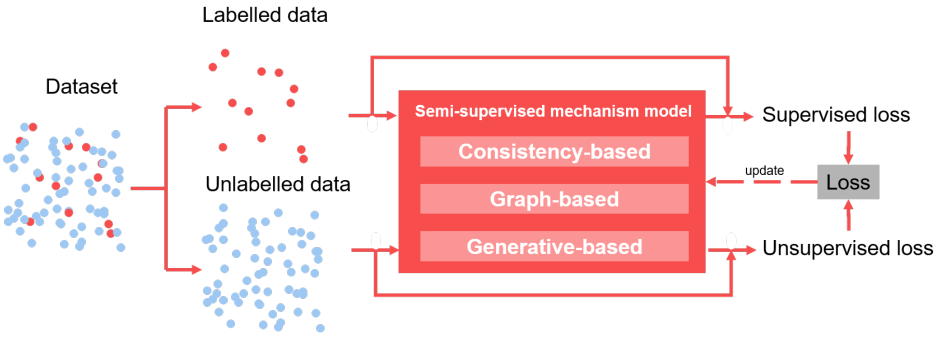

Figure 14. Semi-supervised mechanism-based methods.

Consistency-based methods take advantage of the consistency of model outputs after perturbations [146]; therefore, consistency regularization can be applied for unlabeled data. Consistency constraint can be either imposed between the predictions from perturbed inputs from the same sample, for instance, the π model [147], or between the predictions from two models with the same architecture, such as MeanTeacher [148]. Thanks to the perturbations in consistency-based methods, model generalization can be enhanced [149]. In terms of applications in manufacturing, depending on the output values consistency-based methods can be used in the Know-what and Know-when stages. For instance, consistency-based methods can be utilized in quality monitoring based on images (Know-what, product level).

Graph-based methods aim to establish a graph from a dataset by denoting each data point as a node, with the edge connecting two nodes representing the similarity between them. Label propagation is then performed on the established graph, with the information from labeled data used to infer the labels of the unlabeled data. Graph-based methods result in the connected nodes being closer in the feature space, while disconnected nodes repel each other. Therefore, graph-based methods can be used to address the problem of poor class separation due to intra-class variations and inter-class similarities [18]. Consequently, graph-based methods can be potentially useful for defect classification [18] (Know-what, product level) or machine health state monitoring [150] (Know-what, machine level) where there are problems with insufficient label information or poor class separation. However, it should be noted that graph-based methods are normally transductive methods, meaning that the constructed graph is only valid for the trained data and rebuilding the graph is necessary when it comes to new data. Typical examples of graph-based methods include Graph Neural Networks (GNNs) [151] and Graph Convolution Networks (GCNs) [152].

The main point of generative-based methods is to learn patterns from a dataset and to model data distributions, allowing the model to be used to generate new samples. Then during training, the model can be updated using the combination of the supervised loss (for existing data with labels) and unsupervised loss (for synthetic data). An inherent advantage of generative-based methods is that the labeled data can be enriched by a trained model which has learned the data distribution. Therefore, generative-based methods are well-suited for situations where it is difficult to collect labeled data, such as process fault detection [153] (Know-what, process level) and anomaly detection in machinery [154] (Know-what, machine level). Examples include the semi-supervised GAN series (SS-GANs), such as Categorical Generative Adversarial Network (CatGAN) [155], Improved GAN [156], and semi-supervised VAEs (SS-VAEs) [157].

Table A3 lists semi-supervised applications in manufacturing taken from the selected documents in Section 2.2.

4.4. Reinforcement Learning Methods

Reinforcement Learning (RL) algorithms consist of two elements, namely, an ***agent*** acting within an ***environment*** (see Figure 15). The agent is acting, and is therefore subject to the desired learning process by directly interacting with and manipulating the environment. Based on [158], the procedure of a learning cycle is as follows: first, the agent is presented with an observation of the environment state $s_t \in \mathbb{S}$; then, based on this observation (along with internal decision making), the selection of an action $a_t \in \mathbb{A}$. $\mathbb{S}$ refers to the state space, that is, the set of possible observations that could occur in the environment. The observation has to provide sufficient information on the current environment or system state in order for the agent to select actions in an ideal way to solve the control problem. For selecting the action, $\mathbb{A}$ refers to the action space, that is, the set of possible actions chosen by the agent. After a_t is performed (in a given state s_t), the environment moves to the resulting state s_{t+1} and the agent receives a reward r_{t+1}. Then, the reinforcement learning cycle continues to iterate as shown in Figure 15. The agent aims to maximize the (discounted) long-term cumulative reward by improving the selection of actions towards an optimum. In other words, the RL agent wants to learn an optimal control policy for the environment.

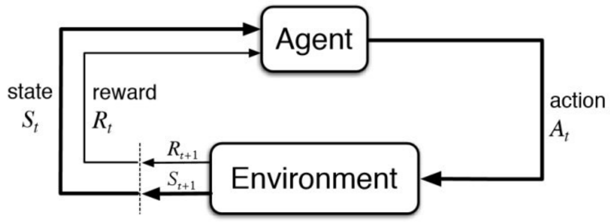

Figure 15. Overview of the Reinforcement Learning approach based on [158].

In general, RL approaches can be split into model-based, i.e., the agent has an internal model of how the environment works, and model-free. The latter is most common thanks to the advent of deep learning, and simplifies application, as feature selection can be applied. Model-free approaches themselves can be divided into short value-based or policy-based approaches by their approach to storing state-action value pairs, which are used to select the action for optimal value return; the latter directly optimize the action selection policy. In contrast to the other machine learning techniques, RL does not require large dataset, only a clearly specified environment. Typically, an RL agent is trained on a simulation or digital twin model [159]; after successful training, it can be implemented on the Know-how level for its original purpose. Otherwise, the agent starts with random non-optimal actions, leading to undesired system behavior.

Considering the aim of achieving the Know-how level for autonomous control in processes, machines, or systems, RL is extremely important for applications in future production. In addition, multi-agent RL is becoming of interest to the research community [33], and can even be applied for controlling products [160]. However, RL remains under-exploited in the industrial area, especially in respect to other machine learning techniques [161].

As of now, applied approaches can be summarized as shown in Table A4. Note that the applications reviewed here are implemented in a simulation or digital twin [159], and features are manually crafted from raw data.

5. Challenges and Future Directions

A large number of ML use cases have shown the great potential for addressing complex manufacturing problems, from knowing what is happening to knowing how employ self-adapting or self-optimizing systems. The data-driven mechanisms in ML enable broader applications in different fields as well as at different levels, from individual products to whole systems. However, in spite of the great potential and advantages offered by ML and numerous off-the-shelf ML models, there are critical challenges to overcome before the successful application of ML in manufacturing can be realized. The following demonstrate typical challenges that manufacturing industries might confront during the application and deployment of ML-based solutions, along with corresponding future directions for tackling these challenges (RQ3).

- *Lack of data*. Preparing the data used for ML is not a simple task, as the scale and the quality of data can greatly affect the performance of ML models. The most common challenge involves preparing a large amount of organized input data, and ensuring high-quality labels if labels are needed. Despite manufacturing data becoming increasingly more accessible due to the development of sensors and the Internet of Things, gathering meaningful data is time-consuming and costly in many cases, for example, fault detection and RUL prediction. This issue might be alleviated by the Synthetic Minority Over-sampling Technique (SMOTE) [162]. However, SMOTE cannot capture complex representative data, as it often relies on interpolation [163]. Data augmentation [139,164] or transfer learning [165] may address this problem. The aim of data augmentation is to enlarge dataset by means of transforming data [139], by transforming both data and labels, as with MixUp [166], or by generating synthetic data using generative models [167,168]. On the contrary, instead of focusing on expanding data, transfer learning aims to leverage knowledge from similar external datasets. A typically used method in transfer learning is parameter transfer, where a pretrained model from a similar dataset is employed for initialization [165]. Another situation involving lack of data is that certain data cannot be shared due to data privacy and security issues. In confronting this problem, Federated Learning (FL) [169] might be a potential opportunity to enable model training across multiple decentralized devices while holding local data privately.

- *Limited computing resources*. The high performance of ML models always comes with high computational complexity. In particular, obtaining high accuracy with a neural network requires on millions or even billions of parameters [170]. However, limited computing resources in industries makes it a challenge to deploy heavy ML models in real-time industrial environments. Possible approaches include model compression via pruning and sharing of model parameters [171] and knowledge distillation [172]. Parameter pruning aims to reduce the number of model parameters by removing redundant parameters without any effect on model performance. By contrast, seeking the same goal, knowledge distillation focuses on distilling knowledge from a cumbersome neural network to a lightweight network to allow it to be deployed more easily with limited computing resources.
- *Changing circumstances*. Most ML applications in manufacturing focus only on model development and verification in off-line environments. However, when deploying these models in running production, their performance may be degraded due to changing circumstances, leading to changes in data distribution, that is, drift [173,174]. Therefore, manual model adjustment over time, which is time-consuming, is usually unavoidable [175]. However, this could be addressed in the future by automatic model adaption [174], in which data drifts are automatically detected and handled with less resources.
- *Interpretability of results*. Many expectations have been placed on ML to overcome all types of problems without the need for prior knowledge. In particular, ML models are expected to directly learn higher level knowledge such as Know-when and Know-how, which is difficult for human beings to obtain in manufacturing. However, without the foundations of early-stage knowledge and an understanding of the data, the results inferred from big data by black-box ML models are meaningless and unreliable. For instance, predictions blindly obtained from all data, including both relevant and irrelevant data, might even degrade performance due to the GIGO (garbage in, garbage out) phenomenon [176]. To overcome this problem, future directions within ML development might include incorporating physical models into ML models [177] or obtaining Four-know knowledge successively.
- *Uncertainty of results*. Related to the challenge of interpretability is the challenge of uncertain results. The success of manufacturing depends heavily on the quality of the resulting products. As every manufacturing process has a degree of variability, almost all industrial manufacturers use statistical process control (SPC) to ensure a stable and defined quality of products [178]. A central element of statistical process control is the determination and handling of statistical uncertainty. The uncertainty of ML results often cannot be quantified reliably and efficiently, even with today's state-of-the-art [179–181]. Furthermore, model complexity and severe non-linearity in ML can hinder the evaluation of uncertainty [182]. Although there are promising approaches, e.g., Gaussian mixture models for NN [183,184] and Probabilistic Neural Network (PNN) [184], or the use of Baysian Networks [180], there are several limitations limiting potential applications, such as high computational cost and simplified assumptions [184]. Therefore, future research needs to make progress on the general theory of integrating uncertainty into ML methods to allow manufacturing in order to ensure high quality and stability in production.

To summarize, while ML is a fairly open tool which can be used to handle a variety of problems in manufacturing, it is necessary to have an understanding of the hidden challenges in ML application in order to provide more realistic and robust outcomes. For instance, early in ML application in manufacturing, one might face the problem of lacking data. During the deployment of ML-based solutions, one might confront challenges around integrating the solution into the industrial environment. After deployment, one might encounter the challenge of evaluating ML results on product and process in terms of interpretability and uncertainty. The future directions pointed out in this review can help to address the above-mentioned challenges and ensure reliable improvements in manufacturing contexts.

6. Conclusions

It is fully recognized that ML is playing an increasingly critical role in the digitization of manufacturing industries towards Industry 4.0, leading to improved quality, productivity, and efficiency. This review has paper aimed to address the issue of how ML can improve manufacturing, posing three research questions related to the above issue in the introduction. To address these research questions, we carried out a literature review assessing the state-of-the-art based on 1348 published scientific articles.

To answer RQ1, we first introduced the concepts of the ‘Four-Know’ (Know-what, Know-why, Know-when, Know-how) and ‘Four-Level’ (Product, Process, Machine, System) categories to help formulate ML tasks in manufacturing. By mapping ML use cases into the Four-Know and Four-Level matrix, we provide an understanding of typical ML use cases and their potential benefits for improving manufacturing. To further support RQ1, the identified ML studies were classified using the ’Four-Know’ and ’Four-Level’ perspective to provide an overview of ML publications in manufacturing. The results showed that current ML applications are mainly focused on the product level, in particular in terms of Know-what and Know-when. In addition, considerable growth in Know-how was observed at the process and system levels, which might be correlated to higher demand for adaptability to changes on these levels.

To fill the gap between academic research and manufacturing industries, we provided an actionable pipeline for the implementation of ML solutions by production engineers from ideation through to deployment, thereby answering RQ2. To further explain the ’model training’ step, which is the core stage in the pipeline, a holistic review of ML methods was provided, including supervised, semi-supervised, unsupervised, and reinforcement learning methods along with their typical applications in manufacturing. We hope that this can provide support in method selection for decision-makers considering ML solutions.

Finally, to answer RQ3, we uncovered the current challenges that manufacturing industry is likely to encounter during application and deployment, and provided possible future directions for tackling these challenges as possible developments for ensuring more reliable and robust outcomes in manufacturing.

Author Contributions: Conceptualization, T.C., O.J.J., M.C.M., V.S. and G.F.; methodology, T.C. and S.S.; formal analysis, T.C. and S.S.; writing—original draft preparation, T.C., V.S., S.S., M.C.M., O.J.J. and F.S.; writing—review and editing, T.C., V.S., M.C.M., S.S., O.J.J., M.C., G.F., G.T., J.J.A.M. and F.S.; supervision, M.C., G.F., G.T., J.J.A.M. and F.S.; funding acquisition, G.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by a European Training Network supported by Horizon 2020, grant number 814225.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research work was undertaken in the context of the DIGIMAN4.0 project (“DIGItal MANufacturing Technologies for Zero-defect Industry 4.0 Production”, https://www.digiman4-0.mek.dtu.dk/, accessed on 1 January 2023). DIGIMAN4.0 is a European Training Network supported by Horizon 2020, the EU Framework Programme for Research and Innovation (Project ID: 814225).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Categories of supervised learning applications.

Ref.	Year	Level	Know-What	Know-Why	Know-When	Know-How	Data Type	Method Type	Case	Field
[104]	2018	Product	✓				Image	Kernel	Defect detection	Metallic powder bed fusion
[185]	2018	Product	✓				Tabular	Kernel	Product monitoring	Metal frame process in mobile device manufacturing
[186]	2020	Process	✓	✓	✓		Time series, Image	Kernel	Temperature prediction and potential anomaly detection	Additive manufacturing
[19]	2022	Product, Process	✓	✓			Tabular	Kernel	Fault detection and classification	Semiconductor Etch Equipment
[105]	2022	Product			✓		Tabular	Kernel	Quality prediction	Additive manufacturing
[106]	2022	Machine			✓		Time series	Kernel	Wear prediction	Metal forming
[187]	2022	System		✓	✓		Time series	Kernel	Content prediction	Steel making
[114]	2018	Machine	✓	✓			Time series (Image)	NN	Fault diagnosis	Motor bearning and pump
[188]	2020	System		✓	✓		Image	NN	Cost estimation	
[112]	2020	Product	✓				Image	NN	Defect detection	Battery manufacturing
[189]	2022	Product	✓				Time series	NN	Quality assurance	Fused deposition modeling
[190]	2022	Process	✓				Time series	NN	Process optimization	Wire arc additive manufacturing
[191]	2022	Process			✓	✓	Tabular	NN	Parameter optimization	Laser powder bed fusion
[192]	2022	Process	✓				Image	NN	Object detection	Robotic grasp
[193]	2022	Product	✓				Image	NN, kernel	Defect detection	Roller manufacturing
[194]	2022	Machine			✓		Time series (Image)	NN, kernel	Tool condition monitoring	Machining
[195]	2019	Product			✓		Time series	Tree	Material removal prediction	Robotic grinding
[196]	2022	Product			✓		Image (Tabular)	Tree	Porosity prediction	Powder-bed additive manufacturing
[197]	2019	Product	✓				Image	Probabilistic	Online quality inspection	Powder-bed additive manufacturing
[198]	2018	System				✓	Tabular	Hybrid	Scheduling	Flexible Manufacturing Systems (FMSs)

Table A2. Categories of unsupervised learning applications.

Ref.	Year	Level	Know-What	Know-Why	Know-When	Know-How	Data Type	Method Type	Case	Field
[120]	2021	Machine	✓				Time series	Clustering	Tool Condition clustering	Autonomous manufacturing
[131]	2021	Machine	✓				Time series	Clustering	Tool health monitoring	Machine tool health Monitoring
[132]	2019	Machine	✓	✓			Time series	Clustering	Defect Identification	Manufacturing systems
[199]	2021	Process	✓				Tabular, Time series	Clustering	Condition monitoring	Manufacturing Condition monitoring
[133]	2020	System	✓				Time series	Clustering	Condition monitoring	Manufacturing Condition monitoring
[125]	2018	Product	✓				Image, Text	Autoencoder	Defect Identification	Fabric industry
[119]	2019	Product	✓				Image	Autoencoder	Defect Identification	Automatic Optical Inspection
[200]	2021	Product	✓				Image	Autoencoder	Defect Identification	Printed circuit board manufacturing
[201]	2022	Machine	✓		✓		Tabular, Time series	Autoencoder	Anomaly detection	Steel rolling Process
[126]	2022	Process	✓		✓		Image	Autoencoder	Anomaly detection	Industrial Anomaly detection
[126]	2022	Process	✓		✓		Image, Text	Autoencoder	Anomaly detection	Semi conductor manufacturing
[202]	2022	Machine			✓		Time series	PCA	Predictive maintenance	Fan-motor system
[118]	2022	Machine	✓	✓			Time series	PCA	Anomaly detection	Programmable logic controllers
[121]	2015	Process	✓				Tabular	Association rule	Predictive maintenance	Wooden door manufacturing

Table A3. Categories of semi-supervised learning applications.

Ref.	Year	Level	Know-What	Know-Why	Know-When	Know-How	Data Type	Method Type	Case	Field
[203]	2020	Product	✓				Image	Data augmentation	Quality control	Automated Surface Inspection
[204]	2021	Process	✓				Image	Data augmentation	Measurement in process	Positioning of welding seams
[205]	2019	System			✓		Tabular	Data augmentation	Energy consumption modelling	Steel industry
[206]	2020	Product, System			✓		Time series	Data augmentation	Quality prediction	Continuous-flow manufacturing.
[207]	2021	Machine	✓				Time series	Consistency-based	Predictive quality control	Semiconductor manufacturing
[149]	2020	Product	✓				Image	Consistency-based	Quality monitoring	Metal additive manufacturing
[18]	2021	Product	✓				Image	Graph-based	Quality control	Automated Surface Inspection
[150]	2022	Machine	✓	✓			Time series	Graph-based	Machine health state diagnosis	Manipulator
[208]	2022	Machine		✓	✓		Tabular	Graph-based	Predict tool tip dynamics	Machine tool
[209]	2021	Product			✓		Image	Generative-based	Assessing manufacturability of cellular structures	Direct metal laser sintering process
[210]	2019	Product	✓				Time series	Generative-based	Quality inferred from process	laser powder-bed fusion
[211]	2020	Product	✓				Image	Generative-based	Quality diagnosis	Wafer fabrication
[212]	2021	Product	✓				Image	Generative-based	Quality control	Automated Surface Inspection
[213]	2020	Machine			✓		Time series	Generative-based	Remaining useful life prognostics	Turbofan engine and rolling bearing
[214]	2021	Machine	✓	✓			Tabular	Generative-based	Machine condition monitoring	Vacuum system in styrene petrochemical plant
[153]	2021	Machine	✓	✓			Time series	Generative-based	Anomaly detection for predictive maintenance	Press machine
[154]	2022	Process	✓				Time series (image)	Generative-based	Process fault detection	Die casting process

Table A4. Categories of reinforcement learning applications.

Ref.	Year	Level	Know-What	Know-Why	Know-When	Know-How	Data Type	Method Type	Case	Field
[32]	2021	Process	✓			✓	Tabular	Value-based	Quality control	Statistical Process Control
[215]	2022	System			✓	✓	Tabular	Value-based	Scheduling	Semiconductor fab
[216]	2021	System				✓	Tabular	Value-based	Throughput control	Flow shop
[217]	2021	Machine				✓	Tabular	Value-based	Scheduling & Maintenance	Multi-state single machine
[34]	2020	System				✓	Tabular	Value-based	Quality Control & Maintenance	Production system
[218]	2022	System				✓	Tabular	Value-based	Lead time management	Flow shop
[219]	2020	Process				✓	Tabular	Value-based	Robotic arm control	Soft fabric manufacturing
[220]	2021	System			✓	✓	Tabular	Value-based	Layout planning	Greenfield factories
[221]	2020	Machine				✓	Tabular	Value-based	Maintenance scheduling	Preventive maintenance
[33]	2022	Machine			✓	✓	Tabular	Policy-based	Maintenance scheduling	Parallel machines
[222]	2021	Process				✓	Tabular	Policy-based	Improving efficiency	Automated product disassembly
[223]	2021	System				✓	Tabular	Policy-based	Dispatching	Job shop
[224]	2022	System				✓	Tabular	Policy-based	Scheduling & maintenance	Semiconductor fab
[225]	2022	System				✓	Tabular	Policy-based	Yield optimization	Multi-agent RL
[57]	2022	System				✓	Tabular	Policy-based	Human Worker Control	Flow shop
[59]	2022	System				✓	Tabular	Policy-based	Scheduling & dispatching	Disassembly job shop
[160]	2021	Product				✓	Tabular	Policy-based	Multi-agent production control	Job shop
[226]	2022	Process				✓	Tabular	Both	Parameter optimisation	Manufacturing processes
[227]	2019	Process				✓	Tabular	Both	Online parameter optimisation	Injection molding
[228]	2022	System				✓	Tabular	Both	scheduling	Matrix production system

References

1. Abele, E.; Reinhart, G. *Zukunft der Produktion: Herausforderungen, Forschungsfelder, Chancen*; Hanser: München, Germany, 2011.
2. Zizic, M.C.; Mladineo, M.; Gjeldum, N.; Celent, L. From industry 4.0 towards industry 5.0: A review and analysis of paradigm shift for the people, organization and technology. *Energies* **2022**, *15*, 5221. [CrossRef]
3. Huang, S.; Wang, B.; Li, X.; Zheng, P.; Mourtzis, D.; Wang, L. Industry 5.0 and Society 5.0—Comparison, complementation and co-evolution. *J. Manuf. Syst.* **2022**, *64*, 424–428. [CrossRef]
4. Vukovic, M.; Mazzei, D.; Chessa, S.; Fantoni, G. Digital Twins in Industrial IoT: A survey of the state of the art and of relevant standards. In Proceedings of the 2021 IEEE International Conference on Communications Workshops (ICC Workshops), Montreal, QC, Canada, 14–23 June 2021. [CrossRef]
5. Mourtzis, D.; Fotia, S.; Boli, N.; Vlachou, E. Modelling and quantification of industry 4.0 manufacturing complexity based on information theory: A robotics case study. *Int. J. Prod. Res.* **2019**, *57*, 6908–6921. [CrossRef]
6. Galin, R.; Meshcheryakov, R.; Kamesheva, S.; Samoshina, A. Cobots and the benefits of their implementation in intelligent manufacturing. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *862*, 032075. [CrossRef]
7. May, M.C.; Schmidt, S.; Kuhnle, A.; Stricker, N.; Lanza, G. Product Generation Module: Automated Production Planning for optimized workload and increased efficiency in Matrix Production Systems. *Procedia CIRP* **2020**, *96*, 45–50. [CrossRef]
8. Lu, Y. Industry 4.0: A survey on technologies, applications and open research issues. *J. Ind. Inf. Integr.* **2017**, *6*, 1–10. [CrossRef]
9. Miqueo, A.; Torralba, M.; Yagüe-Fabra, J.A. Lean manual assembly 4.0: A systematic review. *Appl. Sci.* **2020**, *10*, 8555. [CrossRef]
10. Wuest, T.; Weimer, D.; Irgens, C.; Thoben, K.D. Machine learning in manufacturing: Advantages, challenges, and applications. *Prod. Manuf. Res.* **2016**, *4*, 23–45. [CrossRef]
11. Rai, R.; Tiwari, M.K.; Ivanov, D.; Dolgui, A. Machine learning in manufacturing and industry 4.0 applications. *Int. J. Prod. Res.* **2021**, *59*, 4773–4778. [CrossRef]
12. Bertolini, M.; Mezzogori, D.; Neroni, M.; Zammori, F. Machine Learning for industrial applications: A comprehensive literature review. *Expert Syst. Appl.* **2021**, *175*, 114820. [CrossRef]
13. Wang, J.; Ma, Y.; Zhang, L.; Gao, R.X.; Wu, D. Deep learning for smart manufacturing : Methods and applications. *J. Manuf. Syst.* **2018**, *48*, 144–156. [CrossRef]
14. Dogan, A.; Birant, D. Machine learning and data mining in manufacturing. *Expert Syst. Appl.* **2021**, *166*, 114060. [CrossRef]
15. Alshangiti, M.; Sapkota, H.; Murukannaiah, P.K.; Liu, X.; Yu, Q. Why is developing machine learning applications challenging? a study on stack overflow posts. In Proceedings of the 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), Porto de Galinhas, Brazil, 19–20 September 2019; pp. 1–11.
16. Zeller, V.; Hocken, C.; Stich, V. Acatech Industrie 4.0 maturity index—A multidimensional maturity model. In Proceedings of the IFIP International Conference on Advances in Production Management Systems, Seoul, Republic of Korea, 26–30 August 2018; Springer: Cham, Switzerland, 2018; pp. 105–113.
17. Yang, L.; Fan, J.; Huo, B.; Li, E.; Liu, Y. A nondestructive automatic defect detection method with pixelwise segmentation. *Knowl.-Based Syst.* **2022**, *242*, 108338. [CrossRef]
18. Wang, Y.; Gao, L.; Gao, Y.; Li, X. A new graph-based semi-supervised method for surface defect classification. *Robot. Comput. Integr. Manuf.* **2021**, *68*, 102083. [CrossRef]
19. Kim, S.H.; Kim, C.Y.; Seol, D.H.; Choi, J.E.; Hong, S.J. Machine Learning-Based Process-Level Fault Detection and Part-Level Fault Classification in Semiconductor Etch Equipment. *IEEE Trans. Semicond. Manuf.* **2022**, *35*, 174–185. [CrossRef]
20. Peng, S.; Feng, Q.M. Reinforcement learning with Gaussian processes for condition-based maintenance. *Comput. Ind. Eng.* **2021**, *158*, 107321. [CrossRef]
21. Zheng, W.; Liu, Y.; Gao, Z.; Yang, J. Just-in-time semi-supervised soft sensor for quality prediction in industrial rubber mixers. *Chemom. Intell. Lab. Syst.* **2018**, *180*, 36–41. [CrossRef]
22. Kang, P.; Kim, D.; Cho, S. Semi-supervised support vector regression based on self-training with label uncertainty: An application to virtual metrology in semiconductor manufacturing. *Expert Syst. Appl.* **2016**, *51*, 85–106. [CrossRef]
23. Srivastava, A.K.; Patra, P.K.; Jha, R. AHSS applications in Industry 4.0: Determination of optimum processing parameters during coiling process through unsupervised machine learning approach. *Mater. Today Commun.* **2022**, *31*, 103625. [CrossRef]
24. Antomarioni, S.; Ciarapica, F.E.; Bevilacqua, M. Association rules and social network analysis for supporting failure mode effects and criticality analysis : Framework development and insights from an onshore platform. *Saf. Sci.* **2022**, *150*, 105711. [CrossRef]
25. Pan, R.; Li, X.; Chakrabarty, K. Semi-Supervised Root-Cause Analysis with Co-Training for Integrated Systems. In Proceedings of the 2022 IEEE 40th VLSI Test Symposium (VTS), San Diego, CA, USA, 25–27 April 2022. [CrossRef]
26. Chen, R.; Lu, Y.; Witherell, P.; Simpson, T.W.; Kumara, S.; Yang, H. Ontology-Driven Learning of Bayesian Network for Causal Inference and Quality Assurance in Additive Manufacturing. *IEEE Robot. Autom. Lett.* **2021**, *6*, 6032–6038. [CrossRef]
27. Sikder, S.; Mukherjee, I.; Panja, S.C. A synergistic Mahalanobis–Taguchi system and support vector regression based predictive multivariate manufacturing process quality control approach. *J. Manuf. Syst.* **2020**, *57*, 323–337. [CrossRef]
28. Cerquitelli, T.; Ventura, F.; Apiletti, D.; Baralis, E.; Macii, E.; Poncino, M. Enhancing manufacturing intelligence through an unsupervised data-driven methodology for cyclic industrial processes. *Expert Syst. Appl.* **2021**, *182*, 115269. [CrossRef]
29. Kolokas, N.; Vafeiadis, T.; Ioannidis, D.; Tzovaras, D. A generic fault prognostics algorithm for manufacturing industries using unsupervised machine learning classifiers. *Simul. Model. Pract. Theory* **2020**, *103*, 102109. [CrossRef]

30. Verstraete, D.; Droguett, E.; Modarres, M. A deep adversarial approach based on multisensor fusion for remaining useful life prognostics. In Proceedings of the 29th European Safety and Reliability Conference (ESREL 2019), Hannover, Germany, 22–26 September 2020; pp. 1072–1077. [CrossRef]
31. Wu, D.; Jennings, C.; Terpenny, J.; Gao, R.X.; Kumara, S. A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests. *J. Manuf. Sci. Eng. Trans. ASME* **2017**, *139*, 071018. [CrossRef]
32. Viharos, Z.J.; Jakab, R. Reinforcement Learning for Statistical Process Control in Manufacturing. *Meas. J. Int. Meas. Confed.* **2021**, *182*, 109616. [CrossRef]
33. Luis, M.; Rodríguez, R.; Kubler, S.; Giorgio, A.D.; Cordy, M.; Robert, J.; Le, Y. Multi-agent deep reinforcement learning based Predictive Maintenance on parallel machines. *Robot. Comput. Integr. Manuf.* **2022**, *78*, 102406.
34. Paraschos, P.D.; Koulinas, G.K.; Koulouriotis, D.E. Reinforcement learning for combined production-maintenance and quality control of a manufacturing system with deterioration failures. *J. Manuf. Syst.* **2020**, *56*, 470–483. [CrossRef]
35. Liu, Y.H.; Huang, H.P.; Lin, Y.S. Dynamic scheduling of flexible manufacturing system using support vector machines. In Proceedings of the 2005 IEEE Conference on Automation Science and Engineering, IEEE-CASE 2005, Edmonton, AB, Canada, 1–2 August 2005; Volume 2005, pp. 387–392. [CrossRef]
36. Zhou, G.; Chen, Z.; Zhang, C.; Chang, F. An adaptive ensemble deep forest based dynamic scheduling strategy for low carbon flexible job shop under recessive disturbance. *J. Clean. Prod.* **2022**, *337*, 130541. [CrossRef]
37. de la Rosa, F.L.; Gómez-Sirvent, J.L.; Sánchez-Reolid, R.; Morales, R.; Fernández-Caballero, A. Geometric transformation-based data augmentation on defect classification of segmented images of semiconductor materials using a ResNet50 convolutional neural network. *Expert Syst. Appl.* **2022**, *206*, 117731. [CrossRef]
38. Krahe, C.; Marinov, M.; Schmutz, T.; Hermann, Y.; Bonny, M.; May, M.; Lanza, G. AI based geometric similarity search supporting component reuse in engineering design. *Procedia CIRP* **2022**, *109*, 275–280. [CrossRef]
39. Onler, R.; Koca, A.S.; Kirim, B.; Soylemez, E. Multi-objective optimization of binder jet additive manufacturing of Co-Cr-Mo using machine learning. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 1091–1108. [CrossRef]
40. Jadidi, A.; Mi, Y.; Sikström, F.; Nilsen, M.; Ancona, A. Beam Offset Detection in Laser Stake Welding of Tee Joints Using Machine Learning and Spectrometer Measurements. *Sensors* **2022**, *22*, 3881. [CrossRef]
41. Sanchez, S.; Rengasamy, D.; Hyde, C.J.; Figueredo, G.P.; Rothwell, B. Machine learning to determine the main factors affecting creep rates in laser powder bed fusion. *J. Intell. Manuf.* **2021**, *32*, 2353–2373. [CrossRef]
42. Verma, S.; Misra, J.P.; Popli, D. Modeling of friction stir welding of aviation grade aluminium alloy using machine learning approaches. *Int. J. Model. Simul.* **2022**, *42*, 1–8. [CrossRef]
43. Gerling, A.; Ziekow, H.; Hess, A.; Schreier, U.; Seiffer, C.; Abdeslam, D.O. Comparison of algorithms for error prediction in manufacturing with automl and a cost-based metric. *J. Intell. Manuf.* **2022**, *33*, 555–573. [CrossRef]
44. Akbari, P.; Ogoke, F.; Kao, N.Y.; Meidani, K.; Yeh, C.Y.; Lee, W.; Farimani, A.B. MeltpoolNet: Melt pool characteristic prediction in Metal Additive Manufacturing using machine learning. *Addit. Manuf.* **2022**, *55*, 102817. [CrossRef]
45. Dittrich, M.A.; Uhlich, F.; Denkena, B. Self-optimizing tool path generation for 5-axis machining processes. *CIRP J. Manuf. Sci. Technol.* **2019**, *24*, 49–54. [CrossRef]
46. Xi, Z. Model predictive control of melt pool size for the laser powder bed fusion process under process uncertainty. *ASCE-ASME J. Risk Uncertain. Eng. Syst. Part B Mech. Eng.* **2022**, *8*, 011103. [CrossRef]
47. Li, X.; Liu, X.; Yue, C.; Liu, S.; Zhang, B.; Li, R.; Liang, S.Y.; Wang, L. A data-driven approach for tool wear recognition and quantitative prediction based on radar map feature fusion. *Measurement* **2021**, *185*, 110072. [CrossRef]
48. Xia, B.; Wang, K.; Xu, A.; Zeng, P.; Yang, N.; Li, B. Intelligent Fault Diagnosis for Bearings of Industrial Robot Joints Under Varying Working Conditions Based on Deep Adversarial Domain Adaptation. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–13. [CrossRef]
49. May, M.C.; Neidhöfer, J.; Körner, T.; Schäfer, L.; Lanza, G. Applying Natural Language Processing in Manufacturing. *Procedia CIRP* **2022**, *115*, 184–189. [CrossRef]
50. Xu, X.; Li, X.; Ming, W.; Chen, M. A novel multi-scale CNN and attention mechanism method with multi-sensor signal for remaining useful life prediction. *Comput. Ind. Eng.* **2022**, *169*, 108204. [CrossRef]
51. Shah, M.; Vakharia, V.; Chaudhari, R.; Vora, J.; Pimenov, D.Y.; Giasin, K. Tool wear prediction in face milling of stainless steel using singular generative adversarial network and LSTM deep learning models. *Int. J. Adv. Manuf. Technol.* **2022**, *121*, 723–736. [CrossRef]
52. Verl, A.; Steinle, L. Adaptive compensation of the transmission errors in rack-and-pinion drives. *CIRP Ann.* **2022**, *71*, 345–348. [CrossRef]
53. Frigerio, N.; Cornaggia, C.F.; Matta, A. An adaptive policy for on-line Energy-Efficient Control of machine tools under throughput constraint. *J. Clean. Prod.* **2021**, *287*, 125367. [CrossRef]
54. Bozcan, I.; Korndorfer, C.; Madsen, M.W.; Kayacan, E. Score-Based Anomaly Detection for Smart Manufacturing Systems. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 5233–5242. [CrossRef]
55. Bokrantz, J.; Skoogh, A.; Nawcki, M.; Ito, A.; Hagstr, M.; Gandhi, K.; Bergsj, D. Improved root cause analysis supporting resilient production systems. *J. Manuf. Syst.* **2022**, *64*, 468–478. [CrossRef]
56. Long, T.; Li, Y.; Chen, J. Productivity prediction in aircraft final assembly lines: Comparisons and insights in different productivity ranges. *J. Manuf. Syst.* **2022**, *62*, 377–389. [CrossRef]

57. Overbeck, L.; Hugues, A.; May, M.C.; Kuhnle, A.; Lanza, G. Reinforcement Learning Based Production Control of Semi-automated Manufacturing Systems. *Procedia CIRP* **2021**, *103*, 170–175. [CrossRef]
58. May, M.C.; Behnen, L.; Holzer, A.; Kuhnle, A.; Lanza, G. Multi-variate time-series for time constraint adherence prediction in complex job shops. *Procedia CIRP* **2021**, *103*, 55–60. [CrossRef]
59. Wurster, M.; Michel, M.; May, M.C.; Kuhnle, A.; Stricker, N.; Lanza, G. Modelling and condition-based control of a flexible and hybrid disassembly system with manual and autonomous workstations using reinforcement learning. *J. Intell. Manuf.* **2022**, *33*, 575–591. [CrossRef]
60. Liberati, A.; Altman, D.G.; Tetzlaff, J.; Mulrow, C.; Gøtzsche, P.C.; Ioannidis, J.P.; Clarke, M.; Devereaux, P.J.; Kleijnen, J.; Moher, D. The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: Explanation and elaboration. *J. Clin. Epidemiol.* **2009**, *62*, e1–e34. [CrossRef]
61. Sampath, V.; Maurtua, I.; Aguilar Martín, J.J.; Gutierrez, A. A survey on generative adversarial networks for imbalance problems in computer vision tasks. *J. Big Data* **2021**, *8*, 27. [CrossRef]
62. Polyzotis, N.; Roy, S.; Whang, S.E.; Zinkevich, M. Data lifecycle challenges in production machine learning: A survey. *ACM Sigmod Rec.* **2018**, *47*, 17–28. [CrossRef]
63. Wang, Z.; Oates, T. Imaging time-series to improve classification and imputation. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
64. Lee, G.; Flowers, M.; Dyer, M. Learning distributed representations of conceptual knowledge. In Proceedings of the International 1989 Joint Conference on Neural Networks, Washington, DC, USA, 18–22 June 1989. [CrossRef]
65. Zhu, Y.; Brettin, T.; Xia, F.; Partin, A.; Shukla, M.; Yoo, H.; Evrard, Y.A.; Doroshow, J.H.; Stevens, R.L. Converting tabular data into images for deep learning with convolutional neural networks. *Sci. Rep.* **2021**, *11*, 11325. [CrossRef]
66. Sharma, A.; Vans, E.; Shigemizu, D.; Boroevich, K.A.; Tsunoda, T. DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture. *Sci. Rep.* **2019**, *9*, 11399. [CrossRef]
67. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
68. Nanni, L.; Ghidoni, S.; Brahnam, S. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognit.* **2017**, *71*, 158–172. [CrossRef]
69. Alkinani, M.H.; Khan, W.Z.; Arshad, Q.; Raza, M. HSDDD: A Hybrid Scheme for the Detection of Distracted Driving through Fusion of Deep Learning and Handcrafted Features. *Sensors* **2022**, *22*, 1864. [CrossRef]
70. Chen, Z.; Zhang, L.; Cao, Z.; Guo, J. Distilling the Knowledge from Handcrafted Features for Human Activity Recognition. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4334–4342. [CrossRef]
71. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6.
72. Pearson, K. LIII. On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **1901**, *2*, 559–572. [CrossRef]
73. Comon, P. Independent component analysis, a new concept? *Signal Process.* **1994**, *36*, 287–314. [CrossRef]
74. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
75. Mikolov, T.; Karafiát, M.; Burget, L.; Cernockỳ, J.; Khudanpur, S. Recurrent neural network based language model. In *Interspeech*; Makuhari: Chiba-city, Japan, 2010; Volume 2, pp. 1045–1048.
76. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [CrossRef]
77. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
78. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 886–893.
79. Shensa, M.J. The discrete wavelet transform: Wedding the a trous and Mallat algorithms. *IEEE Trans. Signal Process.* **1992**, *40*, 2464–2482. [CrossRef]
80. Gröchenig, K. The short-time Fourier transform. In *Foundations of Time-Frequency Analysis*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2001; pp. 37–58.
81. Harris, Z.S. Distributional structure. *Word* **1954**, *10*, 146–162. [CrossRef]
82. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
83. Liu, D.; Kong, H.; Luo, X.; Liu, W.; Subramaniam, R. Bringing AI to edge: From deep learning's perspective. *Neurocomputing* **2021**, *485*, 297–320. [CrossRef]
84. Gray, R.M.; Neuhoff, D.L. Quantization. *IEEE Trans. Inf. Theory* **1998**, *44*, 2325–2383. [CrossRef]
85. Sampath, V.; Maurtua, I.; Aguilar Martín, J.J.; Iriondo, A.; Lluvia, I.; Rivera, A. Vision Transformer based knowledge distillation for fasteners defect detection. In Proceedings of the 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET), Prague, Czech Republic, 20–22 July 2022; pp. 1–6.
86. Shelden, R. Decision Tree. *Chem. Eng. Prog.* **1970**, *66*, 8.
87. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]

88. Freund, Y.; Schapire, R.E.; others. Experiments with a new boosting algorithm.. In Proceedings of the Thirteenth International Conference on International Conference on Machine Learning (ICML'96), Bari, Italy, 3–6 July 1996; Volume 96, pp. 148–156.
89. Ho, T.K. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–16 August 1995; Volume 1, pp. 278–282.
90. Chen, T.; Guestrin, C. XGBoost. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016. [CrossRef]
91. Choi, S.; Battulga, L.; Nasridinov, A.; Yoo, K.H. A decision tree approach for identifying defective products in the manufacturing process. *Int. J. Contents* **2017**, *13*, 57–65.
92. Sugumaran, V.; Muralidharan, V.; Ramachandran, K. Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing. *Mech. Syst. Signal Process.* **2007**, *21*, 930–942. [CrossRef]
93. Hung, Y.H. Improved ensemble-learning algorithm for predictive maintenance in the manufacturing process. *Appl. Sci.* **2021**, *11*, 6832. [CrossRef]
94. Močkus, J. On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference Novosibirsk, Novosibirsk, Russia, 1–7 July 1974*; Marchuk, G.I., Ed.; Springer: Berlin/Heidelberg, Germany, 1975; pp. 400–404.
95. Baum, L.E.; Petrie, T. Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Stat.* **1966**, *37*, 1554–1563. [CrossRef]
96. Papananias, M.; McLeay, T.E.; Mahfouf, M.; Kadirkamanathan, V. A Bayesian framework to estimate part quality and associated uncertainties in multistage manufacturing. *Comput. Ind.* **2019**, *105*, 35–47. [CrossRef]
97. Patange, A.D.; Jegadeeshwaran, R. Application of bayesian family classifiers for cutting tool inserts health monitoring on CNC milling. *Int. J. Progn. Health Manag.* **2020**, *11*. [CrossRef]
98. Pandita, P.; Ghosh, S.; Gupta, V.K.; Meshkov, A.; Wang, L. Application of Deep Transfer Learning and Uncertainty Quantification for Process Identification in Powder Bed Fusion. *ASME J. Risk Uncertain. Part B Mech. Eng.* **2022**, *8*, 011106. [CrossRef]
99. Farahani, A.; Tohidi, H.; Shoja, A. An integrated optimization of quality control chart parameters and preventive maintenance using Markov chain. *Adv. Prod. Eng. Manag.* **2019**, *14*, 5–14. [CrossRef]
100. El Haoud, N.; Bachiri, Z. Stochastic artificial intelligence benefits and supply chain management inventory prediction. In Proceedings of the 2019 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA), Paris, France, 12–14 June 2019; pp. 1–5.
101. Feng, M.; Li, Y. Predictive Maintenance Decision Making Based on Reinforcement Learning in Multistage Production Systems. *IEEE Access* **2022**, *10*, 18910–18921. [CrossRef]
102. Sobaszek, Ł.; Gola, A.; Kozłowski, E. Predictive scheduling with Markov chains and ARIMA models. *Appl. Sci.* **2020**, *10*, 6121. [CrossRef]
103. Hofmann, T.; Schölkopf, B.; Smola, A.J. Kernel methods in machine learning. *Ann. Stat.* **2008**, *36*, 1171–1220. [CrossRef]
104. Gobert, C.; Reutzel, E.W.; Petrich, J.; Nassar, A.R.; Phoha, S. Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging. *Addit. Manuf.* **2018**, *21*, 517–528. [CrossRef]
105. McGregor, D.J.; Bimrose, M.V.; Shao, C.; Tawfick, S.; King, W.P. Using machine learning to predict dimensions and qualify diverse part designs across multiple additive machines and materials. *Addit. Manuf.* **2022**, *55*, 102848. [CrossRef]
106. Kubik, C.; Knauer, S.M.; Groche, P. Smart sheet metal forming: Importance of data acquisition, preprocessing and transformation on the performance of a multiclass support vector machine for predicting wear states during blanking. *J. Intell. Manuf.* **2022**, *33*, 259–282. [CrossRef]
107. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
108. Mika, S.; Ratsch, G.; Weston, J.; Scholkopf, B.; Mullers, K. Fisher discriminant analysis with kernels. In Proceedings of the Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (Cat. No.98TH8468), Madison, WI, USA, 25 August 1999; pp. 41–48. [CrossRef]
109. Fukushima, K.; Miyake, S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and Cooperation in Neural Nets*; Springer: Berlin/Heidelberg, Germany, 1982; pp. 267–285.
110. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
111. Hinton, G.E. Deep belief networks. *Scholarpedia* **2009**, *4*, 5947. [CrossRef]
112. Badmos, O.; Kopp, A.; Bernthaler, T.; Schneider, G. Image-based defect detection in lithium-ion battery electrode using convolutional neural networks. *J. Intell. Manuf.* **2020**, *31*, 885–897. [CrossRef]
113. Ho, S.; Zhang, W.; Young, W.; Buchholz, M.; Al Jufout, S.; Dajani, K.; Bian, L.; Mozumdar, M. DLAM: Deep Learning Based Real-Time Porosity Prediction for Additive Manufacturing Using Thermal Images of the Melt Pool. *IEEE Access* **2021**, *9*, 115100–115114. [CrossRef]
114. Wen, L.; Li, X.; Gao, L.; Zhang, Y. A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Trans. Ind. Electron.* **2017**, *65*, 5990–5998. [CrossRef]
115. Al-Dulaimi, A.; Zabihi, S.; Asif, A.; Mohammadi, A. A multimodal and hybrid deep neural network model for remaining useful life estimation. *Comput. Ind.* **2019**, *108*, 186–196. [CrossRef]

116. Huang, J.; Segura, L.J.; Wang, T.; Zhao, G.; Sun, H.; Zhou, C. Unsupervised learning for the droplet evolution prediction and process dynamics understanding in inkjet printing. *Addit. Manuf.* **2020**, *35*, 101197. [CrossRef]
117. Huang, J.; Chang, Q.; Arinez, J. Product completion time prediction using a hybrid approach combining deep learning and system model. *J. Manuf. Syst.* **2020**, *57*, 311–322. [CrossRef]
118. Cohen, J.; Jiang, B.; Ni, J. Machine Learning for Diagnosis of Event Synchronization Faults in Discrete Manufacturing Systems. *J. Manuf. Sci. Eng.* **2022**, *144*, 071006. [CrossRef]
119. Mujeeb, A.; Dai, W.; Erdt, M.; Sourin, A. One class based feature learning approach for defect detection using deep autoencoders. *Adv. Eng. Inform.* **2019**, *42*, 100933. [CrossRef]
120. Kasim, N.; Nuawi, M.; Ghani, J.; Rizal, M.; Ngatiman, N.; Haron, C. Enhancing Clustering Algorithm with Initial Centroids in Tool Wear Region Recognition. *Int. J. Precis. Eng. Manuf.* **2021**, *22*, 843–863. [CrossRef]
121. Djatna, T.; Alitu, I.M. An application of association rule mining in total productive maintenance strategy: An analysis and modelling in wooden door manufacturing industry. *Procedia Manuf.* **2015**, *4*, 336–343. [CrossRef]
122. Chiang, L.H.; Colegrove, L.F. Industrial implementation of on-line multivariate quality control. *Chemom. Intell. Lab. Syst.* **2007**, *88*, 143–153. [CrossRef]
123. You, D.; Gao, X.; Katayama, S. WPD-PCA-based laser welding process monitoring and defects diagnosis by using FNN and SVM. *IEEE Trans. Ind. Electron.* **2014**, *62*, 628–636. [CrossRef]
124. Moshat, S.; Datta, S.; Bandyopadhyay, A.; Pal, P. Optimization of CNC end milling process parameters using PCA-based Taguchi method. *Int. J. Eng. Sci. Technol.* **2010**, *2*, 95–102. [CrossRef]
125. Mei, S.; Wang, Y.; Wen, G. Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model. *Sensors* **2018**, *18*, 1064. [CrossRef] [PubMed]
126. Maggipinto, M.; Beghi, A.; Susto, G.A. A Deep Convolutional Autoencoder-Based Approach for Anomaly Detection With Industrial, Non-Images, 2-Dimensional Data: A Semiconductor Manufacturing Case Study. *IEEE Trans. Autom. Sci. Eng.* **2022**. [CrossRef]
127. Yang, Z.; Gjorgjevikj, D.; Long, J.; Zi, Y.; Zhang, S.; Li, C. Sparse autoencoder-based multi-head deep neural networks for machinery fault diagnostics with detection of novelties. *Chin. J. Mech. Eng.* **2021**, *34*, 54. [CrossRef]
128. Cheng, R.C.; Chen, K.S. Ball bearing multiple failure diagnosis using feature-selected autoencoder model. *Int. J. Adv. Manuf. Technol.* **2022**, *120*, 4803–4819. [CrossRef]
129. Ramamurthy, M.; Robinson, Y.H.; Vimal, S.; Suresh, A. Auto encoder based dimensionality reduction and classification using convolutional neural networks for hyperspectral images. *Microprocess. Microsyst.* **2020**, *79*, 103280. [CrossRef]
130. Angelopoulos, A.; Michailidis, E.T.; Nomikos, N.; Trakadas, P.; Hatziefremidis, A.; Voliotis, S.; Zahariadis, T. Tackling faults in the industry 4.0 era—A survey of machine-learning solutions and key aspects. *Sensors* **2019**, *20*, 109. [CrossRef]
131. de Lima, M.J.; Crovato, C.D.P.; Mejia, R.I.G.; da Rosa Righi, R.; de Oliveira Ramos, G.; da Costa, C.A.; Pesenti, G. HealthMon: An approach for monitoring machines degradation using time-series decomposition, clustering, and metaheuristics. *Comput. Ind. Eng.* **2021**, *162*, 107709. [CrossRef]
132. Song, W.; Wen, L.; Gao, L.; Li, X. Unsupervised fault diagnosis method based on iterative multi-manifold spectral clustering. *IET Collab. Intell. Manuf.* **2019**, *1*, 48–55. [CrossRef]
133. Subramaniyan, M.; Skoogh, A.; Muhammad, A.S.; Bokrantz, J.; Johansson, B.; Roser, C. A generic hierarchical clustering approach for detecting bottlenecks in manufacturing. *J. Manuf. Syst.* **2020**, *55*, 143–158. [CrossRef]
134. Srinivasan, M.; Moon, Y.B. A comprehensive clustering algorithm for strategic analysis of supply chain networks. *Comput. Ind. Eng.* **1999**, *36*, 615–633. [CrossRef]
135. Das, J.N.; Tiwari, M.K.; Sinha, A.K.; Khanzode, V. Integrated warehouse assignment and carton configuration optimization using deep clustering-based evolutionary algorithms. *Expert Syst. Appl.* **2023**, *212*, 118680. [CrossRef]
136. Stojanovic, L.; Dinic, M.; Stojanovic, N.; Stojadinovic, A. Big-data-driven anomaly detection in industry (4.0): An approach and a case study. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016; pp. 1647–1652.
137. Saldivar, A.A.F.; Goh, C.; Li, Y.; Chen, Y.; Yu, H. Identifying smart design attributes for Industry 4.0 customization using a clustering Genetic Algorithm. In Proceedings of the 2016 22nd International Conference on Automation and Computing (ICAC), Colchester, UK, 7–8 September 2016; pp. 408–414.
138. Chen, W.C.; Tseng, S.S.; Wang, C.Y. A novel manufacturing defect detection method using association rule mining techniques. *Expert Syst. Appl.* **2005**, *29*, 807–815. [CrossRef]
139. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
140. Iwana, B.K.; Uchida, S. An empirical survey of data augmentation for time series classification with neural networks. *PLoS ONE* **2021**, *16*, e0254841. [CrossRef]
141. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
142. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
143. Wong, S.C.; Gatt, A.; Stamatescu, V.; McDonnell, M.D. Understanding data augmentation for classification: When to warp? In Proceedings of the 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Gold Coast, Australia, 30 November–2 December 2016; pp. 1–6.

144. Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; Raffel, C.A. Mixmatch: A holistic approach to semi-supervised learning. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 5049–5059.
145. Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C.A.; Cubuk, E.D.; Kurakin, A.; Li, C.L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 596–608.
146. Yang, X.; Song, Z.; King, I.; Xu, Z. A Survey on Deep Semi-supervised Learning. *arXiv* **2021**, arXiv:2103.00550.
147. Sajjadi, M.; Javanmardi, M.; Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 1171–1179.
148. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1195–1204.
149. Li, X.; Jia, X.; Yang, Q.; Lee, J. Quality analysis in metal additive manufacturing with deep learning. *J. Intell. Manuf.* **2020**, *31*, 2003–2017. [CrossRef]
150. Zhao, B.; Zhang, X.; Zhan, Z.; Wu, Q.; Zhang, H. A Novel Semi-Supervised Graph-Guided Approach for Intelligent Health State Diagnosis of a 3-PRR Planar Parallel Manipulator. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 4786–4797. [CrossRef]
151. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural message passing for quantum chemistry. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1263–1272.
152. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
153. Serradilla, O.; Zugasti, E.; Ramirez de Okariz, J.; Rodriguez, J.; Zurutuza, U. Adaptable and explainable predictive maintenance: Semi-supervised deep learning for anomaly detection and diagnosis in press machine data. *Appl. Sci.* **2021**, *11*, 7376. [CrossRef]
154. Song, J.; Lee, Y.C.; Lee, J. Deep generative model with time series-image encoding for manufacturing fault detection in die casting process. *J. Intell. Manuf.* **2022**, 1–14. [CrossRef]
155. Springenberg, J.T. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv* **2015**, arXiv:1511.06390.
156. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training gans. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 2234–2242.
157. Kingma, D.P.; Mohamed, S.; Jimenez Rezende, D.; Welling, M. Semi-supervised learning with deep generative models. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 3581–3589.
158. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
159. May, M.C.; Overbeck, L.; Wurster, M.; Kuhnle, A.; Lanza, G. Foresighted digital twin for situational agent selection in production control. *Procedia CIRP* **2021**, *99*, 27–32. [CrossRef]
160. May, M.C.; Kiefer, L.; Kuhnle, A.; Stricker, N.; Lanza, G. Decentralized multi-agent production control through economic model bidding for matrix production systems. *Procedia Cirp* **2021**, *96*, 3–8. [CrossRef]
161. Yao, M. Breakthrough Research In Reinforcement Learning From 2019. 2019. Available online: https://www.topbots.com/top-ai-reinforcement-learning-research-papers-2019 (accessed on 1 September 2022).
162. Gao, R.X.; Wang, L.; Helu, M.; Teti, R. Big data analytics for smart factories of the future. *CIRP Ann.* **2020**, *69*, 668–692. [CrossRef]
163. Kozjek, D.; Vrabič, R.; Kralj, D.; Butala, P. Interpretative identification of the faulty conditions in a cyclic manufacturing process. *J. Manuf. Syst.* **2017**, *43*, 214–224. [CrossRef]
164. Wen, Q.; Sun, L.; Yang, F.; Song, X.; Gao, J.; Wang, X.; Xu, H. Time series data augmentation for deep learning: A survey. *arXiv* **2020**, arXiv:2002.12478.
165. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [CrossRef]
166. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv* **2017**, arXiv:1710.09412.
167. Bao, J.; Chen, D.; Wen, F.; Li, H.; Hua, G. CVAE-GAN: Fine-grained image generation through asymmetric training. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2745–2754.
168. Yoon, J.; Jarrett, D.; Van der Schaar, M. Time-series generative adversarial networks. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 5508–5518.
169. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
170. Cheng, Y.; Wang, D.; Zhou, P.; Zhang, T. Model compression and acceleration for deep neural networks: The principles, progress, and challenges. *IEEE Signal Process. Mag.* **2018**, *35*, 126–136. [CrossRef]
171. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 1789–1819. [CrossRef]
172. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
173. Schlimmer, J.C.; Granger, R.H. Incremental learning from noisy data. *Mach. Learn.* **1986**, *1*, 317–354. [CrossRef]
174. Gama, J.; Žliobaitė, I.; Bifet, A.; Pechenizkiy, M.; Bouchachia, A. A survey on concept drift adaptation. *ACM Comput. Surv.* **2014**, *46*, 44. [CrossRef]
175. Baier, L.; Jöhren, F.; Seebacher, S. Challenges in the Deployment and Operation of Machine Learning in Practice. In Proceedings of the ECIS 2019 27th European Conference on Information Systems, Stockholm, Sweden, 8–14 June 2019.
176. Canbek, G. Gaining insights in datasets in the shade of "garbage in, garbage out" rationale: Feature space distribution fitting. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2022**, *12*, e1456. [CrossRef]

177. Moges, T.; Yang, Z.; Jones, K.; Feng, S.; Witherell, P.; Lu, Y. Hybrid modeling approach for melt-pool prediction in laser powder bed fusion additive manufacturing. *J. Comput. Inf. Sci. Eng.* **2021**, *21*, 050902. [CrossRef]
178. Colledani, M., Statistical Process Control. In *CIRP Encyclopedia of Production Engineering*; Laperrière, L., Reinhart, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 1150–1157. [CrossRef]
179. Abdar, M.; Pourpanah, F.; Hussain, S.; Rezazadegan, D.; Liu, L.; Ghavamzadeh, M.; Fieguth, P.; Cao, X.; Khosravi, A.; Acharya, U.R.; et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Inf. Fusion* **2021**, *76*, 243–297. [CrossRef]
180. Yong, B.X.; Brintrup, A. Multi Agent System for Machine Learning Under Uncertainty in Cyber Physical Manufacturing System. In *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future*; Borangiu, T., Trentesaux, D., Leitão, P., Giret Boggino, A., Botti, V., Eds.; Studies in Computational Intelligence; Springer International Publishing: Cham, Switzerland, 2020; Volume 853, pp. 244–257. [CrossRef]
181. Tavazza, F.; DeCost, B.; Choudhary, K. Uncertainty Prediction for Machine Learning Models of Material Properties. *ACS Omega* **2021**, *6*, 32431–32440. [CrossRef]
182. Arkov, V. Uncertainty Estimation in Machine Learning. *arXiv* **2022**. [CrossRef]
183. Zhang, B. Data-Driven Uncertainty Analysis in Neural Networks with Applications to Manufacturing Process Monitoring. Ph.D. Thesis, Purdue University Graduate School, West Lafayette, IN, USA, 2021. [CrossRef]
184. Zhang, B.; Shin, Y.C. A probabilistic neural network for uncertainty prediction with applications to manufacturing process monitoring. *Appl. Soft Comput.* **2022**, *124*, 108995. [CrossRef]
185. Lee, S.; Kim, S.B. Time-adaptive support vector data description for nonstationary process monitoring. *Eng. Appl. Artif. Intell.* **2018**, *68*, 18–31. [CrossRef]
186. Gaikwad, A.; Yavari, R.; Montazeri, M.; Cole, K.; Bian, L.; Rao, P. Toward the digital twin of additive manufacturing: Integrating thermal simulations, sensing, and analytics to detect process faults. *IISE Trans.* **2020**, *52*, 1204–1217. [CrossRef]
187. Zhang, C.J.; Zhang, Y.C.; Han, Y. Industrial cyber-physical system driven intelligent prediction model for converter end carbon content in steelmaking plants. *J. Ind. Inf. Integr.* **2022**, *28*, 100356. [CrossRef]
188. Ning, F.; Shi, Y.; Cai, M.; Xu, W.; Zhang, X. Manufacturing cost estimation based on the machining process and deep-learning method. *J. Manuf. Syst.* **2020**, *56*, 11–22. [CrossRef]
189. Westphal, E.; Seitz, H. Machine learning for the intelligent analysis of 3D printing conditions using environmental sensor data to support quality assurance. *Addit. Manuf.* **2022**, *50*, 102535. [CrossRef]
190. Qin, J.; Wang, Y.; Ding, J.; Williams, S. Optimal droplet transfer mode maintenance for wire+ arc additive manufacturing (WAAM) based on deep learning. *J. Intell. Manuf.* **2022**, *33*, 2179–2191. [CrossRef]
191. Lapointe, S.; Guss, G.; Reese, Z.; Strantza, M.; Matthews, M.; Druzgalski, C. Photodiode-based machine learning for optimization of laser powder bed fusion parameters in complex geometries. *Addit. Manuf.* **2022**, *53*, 102687. [CrossRef]
192. Zhang, T.; Zhang, C.; Hu, T. A robotic grasp detection method based on auto-annotated dataset in disordered manufacturing scenarios. *Robot. Comput. Integr. Manuf.* **2022**, *76*, 102329. [CrossRef]
193. Singh, S.A.; Desai, K. Automated surface defect detection framework using machine vision and convolutional neural networks. *J. Intell. Manuf.* **2022**, 1–17. [CrossRef]
194. Duan, J.; Hu, C.; Zhan, X.; Zhou, H.; Liao, G.; Shi, T. MS-SSPCANet: A powerful deep learning framework for tool wear prediction. *Robot. Comput. Integr. Manuf.* **2022**, *78*, 102391. [CrossRef]
195. Gao, K.; Chen, H.; Zhang, X.; Ren, X.; Chen, J.; Chen, X. A novel material removal prediction method based on acoustic sensing and ensemble XGBoost learning algorithm for robotic belt grinding of Inconel 718. *Int. J. Adv. Manuf. Technol.* **2019**, *105*, 217–232. [CrossRef]
196. Gawade, V.; Singh, V.; Guo, W. Leveraging simulated and empirical data-driven insight to supervised-learning for porosity prediction in laser metal deposition. *J. Manuf. Syst.* **2022**, *62*, 875–885. [CrossRef]
197. Aminzadeh, M.; Kurfess, T.R. Online quality inspection using Bayesian classification in powder-bed additive manufacturing from high-resolution visual camera images. *J. Intell. Manuf.* **2019**, *30*, 2505–2523. [CrossRef]
198. Priore, P.; Ponte, B.; Puente, J.; Gómez, A. Learning-based scheduling of flexible manufacturing systems using ensemble methods. *Comput. Ind. Eng.* **2018**, *126*, 282–291. [CrossRef]
199. Guo, S.; Chen, M.; Abolhassani, A.; Kalamdani, R.; Guo, W.G. Identifying manufacturing operational conditions by physics-based feature extraction and ensemble clustering. *J. Manuf. Syst.* **2021**, *60*, 162–175. [CrossRef]
200. Kim, J.; Ko, J.; Choi, H.; Kim, H. Printed circuit board defect detection using deep learning via a skip-connected convolutional autoencoder. *Sensors* **2021**, *21*, 4968. [CrossRef]
201. Jakubowski, J.; Stanisz, P.; Bobek, S.; Nalepa, G.J. Anomaly Detection in Asset Degradation Process Using Variational Autoencoder and Explanations. *Sensors* **2021**, *22*, 291. [CrossRef]
202. Sarita, K.; Devarapalli, R.; Kumar, S.; Malik, H.; Garcia Marquez, F.P.; Rai, P. Principal component analysis technique for early fault detection. *J. Intell. Fuzzy Syst.* **2022**, *42*, 861–872. [CrossRef]
203. Zheng, X.; Wang, H.; Chen, J.; Kong, Y.; Zheng, S. A generic semi-supervised deep learning-based approach for automated surface inspection. *IEEE Access* **2020**, *8*, 114088–114099. [CrossRef]
204. Zhang, W.; Lang, J. Semi-supervised training for positioning of welding seams. *Sensors* **2021**, *21*, 7309. [CrossRef]

205. Chen, C.; Liu, Y.; Kumar, M.; Qin, J.; Ren, Y. Energy consumption modelling using deep learning embedded semi-supervised learning. *Comput. Ind. Eng.* **2019**, *135*, 757–765. [CrossRef]
206. Jun, J.h.; Chang, T.W.; Jun, S. Quality prediction and yield improvement in process manufacturing based on data analytics. *Processes* **2020**, *8*, 1068. [CrossRef]
207. Shim, J.; Cho, S.; Kum, E.; Jeong, S. Adaptive fault detection framework for recipe transition in semiconductor manufacturing. *Comput. Ind. Eng.* **2021**, *161*, 107632. [CrossRef]
208. Qiu, C.; Li, K.; Li, B.; Mao, X.; He, S.; Hao, C.; Yin, L. Semi-supervised graph convolutional network to predict position-and speed-dependent tool tip dynamics with limited labeled data. *Mech. Syst. Signal Process.* **2022**, *164*, 108225. [CrossRef]
209. Guo, Y.; Lu, W.F.; Fuh, J.Y.H. Semi-supervised deep learning based framework for assessing manufacturability of cellular structures in direct metal laser sintering process. *J. Intell. Manuf.* **2021**, *32*, 347–359. [CrossRef]
210. Okaro, I.A.; Jayasinghe, S.; Sutcliffe, C.; Black, K.; Paoletti, P.; Green, P.L. Automatic fault detection for laser powder-bed fusion using semi-supervised machine learning. *Addit. Manuf.* **2019**, *27*, 42–53. [CrossRef]
211. Lee, H.; Kim, H. Semi-supervised multi-label learning for classification of wafer bin maps with mixed-type defect patterns. *IEEE Trans. Semicond. Manuf.* **2020**, *33*, 653–662. [CrossRef]
212. Liu, J.; Song, K.; Feng, M.; Yan, Y.; Tu, Z.; Zhu, L. Semi-supervised anomaly detection with dual prototypes autoencoder for industrial surface inspection. *Opt. Lasers Eng.* **2021**, *136*, 106324. [CrossRef]
213. Verstraete, D.; Droguett, E.; Modarres, M. A deep adversarial approach based on multi-sensor fusion for semi-supervised remaining useful life prognostics. *Sensors* **2019**, *20*, 176. [CrossRef]
214. Souza, M.L.H.; da Costa, C.A.; de Oliveira Ramos, G.; da Rosa Righi, R. A feature identification method to explain anomalies in condition monitoring. *Comput. Ind.* **2021**, *133*, 103528. [CrossRef]
215. Lee, Y.H.; Lee, S. Deep reinforcement learning based scheduling within production plan in semiconductor fabrication. *Expert Syst. Appl.* **2022**, *191*, 116222. [CrossRef]
216. Marchesano, M.G.; Guizzi, G.; Santillo, L.C.; Vespoli, S. A deep reinforcement learning approach for the throughput control of a flow-shop production system. *IFAC-PapersOnLine* **2021**, *54*, 61–66. [CrossRef]
217. Yang, H.; Li, W.; Wang, B. Joint optimization of preventive maintenance and production scheduling for multi-state production systems based on reinforcement learning. *Reliab. Eng. Syst. Saf.* **2021**, *214*, 107713. [CrossRef]
218. Schneckenreither, M.; Haeussler, S.; Peiró, J. Average reward adjusted deep reinforcement learning for order release planning in manufacturing. *Knowl.-Based Syst.* **2022**, *247*, 108765. [CrossRef]
219. Tsai, Y.T.; Lee, C.H.; Liu, T.Y.; Chang, T.J.; Wang, C.S.; Pawar, S.J.; Huang, P.H.; Huang, J.H. Utilization of a reinforcement learning algorithm for the accurate alignment of a robotic arm in a complete soft fabric shoe tongues automation process. *J. Manuf. Syst.* **2020**, *56*, 501–513. [CrossRef]
220. Klar, M.; Glatt, M.; Aurich, J.C. An implementation of a reinforcement learning based algorithm for factory layout planning. *Manuf. Lett.* **2021**, *30*, 1–4. [CrossRef]
221. Huang, J.; Chang, Q.; Arinez, J. Deep reinforcement learning based preventive maintenance policy for serial production lines. *Expert Syst. Appl.* **2020**, *160*, 113701. [CrossRef]
222. Zhang, H.; Peng, Q.; Zhang, J.; Gu, P. Planning for automatic product assembly using reinforcement learning. *Comput. Ind.* **2021**, *130*, 103471. [CrossRef]
223. Kuhnle, A.; May, M.C.; Schaefer, L.; Lanza, G. Explainable reinforcement learning in production control of job shop manufacturing system. *Int. J. Prod. Res.* **2021**, *60*, 5812–5834. [CrossRef]
224. Valet, A.; Altenmüller, T.; Waschneck, B.; May, M.C.; Kuhnle, A.; Lanza, G. Opportunistic maintenance scheduling with deep reinforcement learning. *J. Manuf. Syst.* **2022**, *64*, 518–534. [CrossRef]
225. Huang, J.; Su, J.; Chang, Q. Graph neural network and multi-agent reinforcement learning for machine-process-system integrated control to optimize production yield. *J. Manuf. Syst.* **2022**, *64*, 81–93. [CrossRef]
226. Zimmerling, C.; Poppe, C.; Stein, O.; Kärger, L. Optimisation of manufacturing process parameters for variable component geometries using reinforcement learning. *Mater. Des.* **2022**, *214*, 110423. [CrossRef]
227. Guo, F.; Zhou, X.; Liu, J.; Zhang, Y.; Li, D.; Zhou, H. A reinforcement learning decision model for online process parameters optimization from offline data in injection molding. *Appl. Soft Comput. J.* **2019**, *85*, 105828. [CrossRef]
228. Hofmann, C.; Liu, X.; May, M.; Lanza, G. Hybrid Monte Carlo tree search based multi-objective scheduling. *Prod. Eng.* **2022**, *17*, 133–144. [CrossRef]