applied sciences

Review

Machine Learning in Smart Manufacturing: Challenges and Solutions

Omar Adalat [1], Nereida Polovina [2] and Savas Konur [3,*]

[1] Department of Computing, Imperial College London, London M15 6BH, UK; o.adalat24@imperial.ac.uk
[2] Business School, Manchester Metropolitan University, Manchester M15 6BX, UK; n.polovina@mmu.ac.uk
[3] School of Computer Science and Digital Technologies, Aston University, Birmingham B4 7ET, UK
* Correspondence: s.konur@aston.ac.uk

Abstract

In this paper, we consider the intertwined challenges that arise for process optimisation within smart manufacturing, reviewing fundamental aspects including concept drift in relation to the dynamic nature of manufacturing, the prevalence of heterogeneous data streams, integration of prior domain knowledge, and trustworthiness and transparency of machine learning models. We investigate the real-world nature of these challenges and the proposed solutions in the literature for each of the aforementioned challenges. Whilst other surveys exist for optimisation in the field of smart manufacturing, there is a lack of a comprehensive review addressing the underlying conceptual issues associated with the deployment and adoption of machine learning in smart manufacturing.

Keywords: smart manufacturing; industrial internet of things; machine learning; industrial informatics; artificial intelligence; heterogeneous data; trustworthy systems

1. Introduction

1.1. Smart Manufacturing

Smart manufacturing involves the intersection of multiple research branches, including, but not limited to, the *Industrial Internet of Things* [1,2], *big data analytics* [3], *machine learning* [4,5], and *cloud manufacturing* [6,7] to meet the high levels of adaptability and iteration required in the modern computer-integrated manufacturing environment. In particular, smart manufacturing involves several relevant capabilities of machine learning: *prediction* [8,9]; *explanation* [10]; and data-driven *control*, where reinforcement learning has been used for hybrid real-time control in smart manufacturing [11], as well as control-based learning grounded in game theoretic principles [12]. Smart manufacturing has been the subject of numerous review papers [13–20], with most focusing on broader, generalised issues. However, specific challenges faced by the manufacturing industry in deploying machine learning in practice are often overlooked. This paper aims to address that gap.

Smart manufacturing is a key component of Industry 4.0, which focuses on the integration of advanced digital technologies to enhance and optimise manufacturing processes [21,22]. It plays a crucial role in advancing sustainability by integrating cutting-edge technologies to optimise production processes, minimise environmental impact, and promote efficient resource use. Smart manufacturing enables real-time monitoring and adaptive decision-making, which significantly reduces energy consumption and waste generation. Predictive maintenance can help ensure machinery operates at peak

Academic Editor: Dimitris Mourtzis

Received: 16 July 2026
Revised: 18 August 2026
Accepted: 25 August 2026
Published: 28 August 2026

efficiency, prevent unnecessary downtime, and extend equipment lifespans, thereby reducing resource depletion [4]. Moreover, the adoption of smart manufacturing can support circular economy principles by improving material and product traceability and facilitating reuse, remanufacturing, and recycling across the product lifecycle [23]. Overall, smart manufacturing not only enhances operational efficiency but also creates a sustainable framework for industries to thrive in an increasingly eco-conscious world.

The paper is outlined as follows: Section 1 provides an introduction to smart manufacturing and outlines the current pressing challenges in the sector. Section 2 focuses on the challenge of concept drift; Section 3 covers heterogeneous data streams; and Section 4 focuses on ensuring systems that are developed are trustworthy for deployment. Each of these sections is accompanied by a discussion section. Section 5 summarises our findings in terms of promising future directions, and we provide our recommendations for practical solutions in dealing with and recognising the grand challenges presented.

1.2. Challenges

In this paper, we will present a number of grand challenges that hinder the adoption of smart manufacturing solutions, accompanied by a number of solutions.

1.2.1. Heterogeneous Data Streams

The Industrial Internet of Things is composed of multiple devices and sensors, all with varying data streams and formats that must be accounted for, at a high data velocity and volume.

- The processes of learning should be seamless in the presence of environmental or process-level changes, both of which are known to cause *concept drift*.
- Decisions must be made in real-time involving multiple resources cooperatively [24].
- High-dimensional and heterogeneous data can often be unreliable at varying rates due to factors such as sensors malfunctioning for a brief period of time, corruption of data due to intermittent transmission errors, or maintenance of equipment, all whilst in the presence of noise, which affects the veracity of the data. Solutions deployed should be robust against any such inconsistencies [25].
- Data from an ever-evolving array of sensors and actuators must be fused, including multimodality [26] of structured and unstructured data (numeric readings, images, video, and audio data sources).

1.2.2. Non-Independent and Non-Identically Distributed Data

Moreover, the data that smart manufacturing deals with may not conform to standard independent and identically distributed assumptions present within machine learning because streaming and time-series observations can be temporally dependent or non-stationary. Therefore, methods that rely on independence or stationarity require their assumptions to be checked. We also need to be equipped to handle the dynamic nature of manufacturing, where processes are non-rigid and unexpected changes may occur, i.e., a non-stationary environment. When changes occur, such as producing a variant [27] of a product or a new successor iteration of the product, we wish to investigate how previous data can be utilised as opposed to starting from scratch, which requires dealing with concept drift.

1.2.3. Decision-Making and Domain Knowledge

Another consideration is how to best enhance the decision-making capabilities of production operatives by providing them with transparent, explainable outputs so the information provided is understood and not ignored.

https://doi.org/10.3390/app16178599

Additionally, it is important to retain domain knowledge accumulated over decades [28], instead of only relearning knowledge when capturing data reactively upon rare cases.

1.2.4. Out of Scope

Some challenges that are largely outside the scope of this particular paper include the following.

- Specific state-of-the-art machine learning architectures for optimisation in a broader lens of smart manufacturing. Our review focuses on challenges of deployment and adoption of such methods.
- The delicate interaction between cyber systems and their physical counterparts, and the synchronisation involved.
- Data storage and warehousing solutions for extract, transform, and load (ETL) pipelines.
- Cybersecurity considerations and best practices for dealing with data that contain trade secrets.

1.3. Contributions and Related Work

This paper provides a detailed and concise view of major challenges at the intersection of machine learning and smart manufacturing and identifies active barriers to ubiquitous adoption. By doing so, it provides a foundation for understanding the challenges that implementers of smart manufacturing systems (such as AI engineers and data science teams) can expect. These challenges may be contextually relevant to other areas as well, although our focus remains on smart manufacturing.

Our review focuses on researching the following question: *What are the main conceptual and operational challenges limiting the deployment and continual use of machine learning solutions in smart manufacturing?* In particular, while other studies in this area are numerous, we emphasise the conceptual issues that practitioners are currently facing and will continue to face when deploying machine learning solutions in complex production settings. This is a narrative review and does not claim systematic or exhaustive coverage.

Review Design

This work uses a challenge-led narrative methodology. The review began from the deployment problems represented by the three themes; for each problem, the general methodological literature was identified and mapped to manufacturing-specific studies and examples. Literature gathering was iterative and purposive, using targeted searches around each challenge and its principal solution families, while recent manufacturing reviews were used to situate the scope of the present work. Sources were retained where they helped define a problem, represented a distinct solution family, demonstrated its use in manufacturing, or exposed a relevant limitation.

Table 1 compares the organising scope of the present work with recent reviews. Whereas application- and algorithm-centred reviews ask which methods have been used for particular manufacturing tasks, this review is organised around problems that affect the deployment and sustained usage of those methods.

Table 1. Comparison with recent reviews.

Study	Primary Scope	Organising Perspective	Distinction from the Present Review
Li et al. [16]	Trustworthy AI in human-centric smart manufacturing	Protection, perception, and participation across human roles and product-lifecycle stages	Examines trustworthiness in depth; the present review places trustworthiness alongside concept drift, heterogeneous data, and domain knowledge as interacting deployment challenges.
Deokar et al. [17]	Machine learning for fused deposition modelling	Algorithms for part-quality prediction, defect detection, geometric accuracy, and material efficiency	Concentrates on one additive-manufacturing process and its applications; the present review is process-independent and organised around cross-cutting deployment problems.
Benhanifia et al. [18]	Predictive maintenance in manufacturing	Technological principles, implementation methods, economic consequences, and operational improvements	Concentrates on predictive maintenance; the present review considers challenges that recur across maintenance, quality, control, and other manufacturing tasks.
Bandhana and Vokřínek [19]	AI-driven manufacturing centred on multi-agent-system and manufacturing-execution-system integration	Enabling technologies, integration standards and architecture, human–machine collaboration, and adoption barriers	Takes system integration as its conceptual anchor; the present review takes the sustained operation of deployed machine learning models as its anchor.
Ramesh et al. [20]	Machine learning approaches across manufacturing applications	Comparison of algorithms, applications, and reported performance	Primarily organises evidence by algorithm and application; the present review organises methods by the operational problem they address after or around deployment.
Present review	Deployment and sustained use of machine learning in smart manufacturing	Concept drift, heterogeneous data streams, domain-knowledge integration, and trustworthy systems	Connects general solution families to four interacting, process-independent operational challenges rather than attempting an exhaustive algorithm or application survey.

2. Concept Drift

Manufacturing is a continuously changing, dynamic process rather than a static one. This presents a challenge due to the duality of continuous data streams and the dynamicity of the underlying processes. When slight unforeseen changes occur from previous modelling or the training dataset used, it is not desirable to completely retrain our model (which is inefficient in terms of time and computation) or, in the worst case, disregard all previous valuable data that is now only slightly off. Procuring a new dataset can be expensive and time-prohibitive, making this more than a minor inconvenience. Dealing with an evolving environment forms the challenge of concept drift.

Let X denote the input or explanatory variables, Y the target or dependent variable, and t the current timestep. General distributional drift is a change in the joint distribution $P_t(X,Y) = P_t(Y \mid X)P_t(X)$ between times [29] such that $\exists\, t : \; P_t(X,Y) \neq P_{t+1}(X,Y)$.

Within this broad definition, *real concept drift* changes the predictive relationship $P_t(Y \mid X)$. *Covariate shift*, also termed virtual drift in this literature, changes $P_t(X)$ while $P_t(Y \mid X)$ remains stable. *Prior-probability shift* changes $P_t(Y)$; the more specific label-shift setting usually assumes that $P_t(X \mid Y)$ remains stable. These changes can occur separately or together, as illustrated in Figure 1.

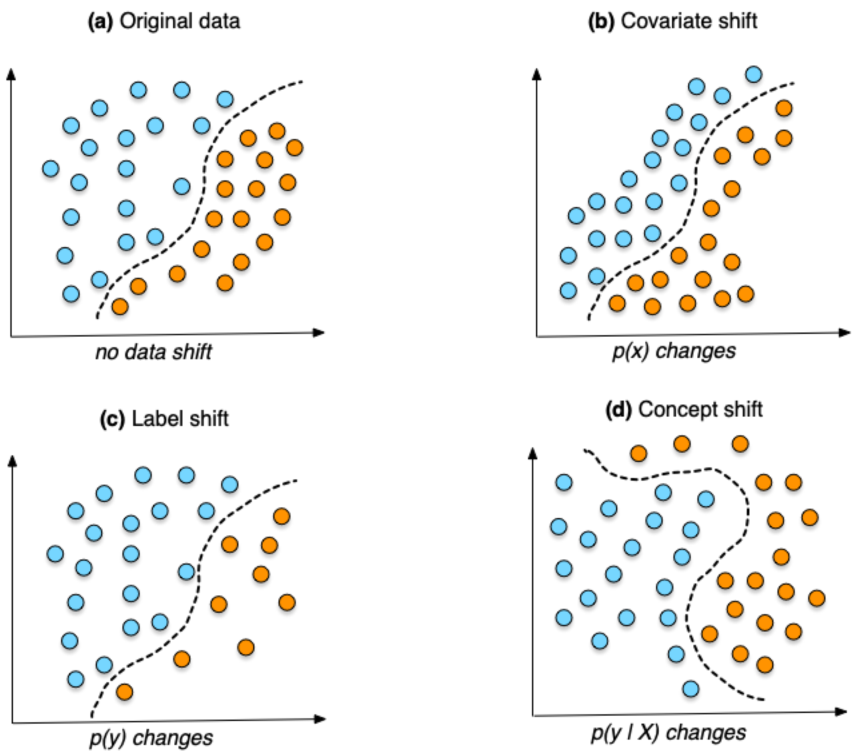

Figure 1. Types of drifts present within datasets. Blue and orange dots represent two distinct data classes (labels).

Concept drift may occur when there is a change in the modelled process, and depending on the machine learning model used, the active model must be either adapted, discarded in favour of new data, or retrained due to concept drift. A key issue presented by concept drift is that limited training data can become further segmented by the presence of drifts. Such drift may arise in smart manufacturing when maintenance schedules, environmental conditions, parts, or input materials change. Changes over time can take the following forms [29], while isolated anomalies must be distinguished from drift:

- *Gradual*: The old and new concepts coexist or alternate during a transition period before the new concept predominates.
- *Incremental*: The distribution passes through a sequence of intermediate concepts. For instance, a sensor may wear and shift its readings progressively until the change stabilises. Recalibration through maintenance may cause the earlier concept to recur.
- *Abrupt*: A sudden change, such as changing input materials, modifying process control parameters/settings or switching sensors which may introduce measurement or calibration discrepancies. In other words, at some time t, there is an abrupt change to a new joint distribution at $t + 1$.
- *Reoccuring*: For example, the material parts may change depending on the supply chain or a slightly different product may be manufactured before switching back. Seasonal production patterns and changing environmental conditions provide further instances. In this case, reoccuring changes may be foreseen and a model operating on the appropriate training data may suffice. Reoccuring drifts may be *cyclical* or *non-cyclical*, where cyclical drifts may have their durations and reoccurence cycles as either fixed or varying time lengths.
- *One-off anomaly*: An isolated random deviation, such as a single erroneous sensor reading, does not constitute concept drift because the underlying concept has not changed.

Following Gama et al. [29], drifts within datasets are characterised in Figure 1, and the temporal relationship of drifts appearing within a dataset is shown in Figure 2. Learning under drift requires relaxing stationarity or identical-distribution assumptions; temporal independence must be assessed separately. There are two objectives that must be met to tackle drifting data streams: *automatically detecting drifts* and *autonomously adapting* to drifts, the latter of which typically (yet not always) involves the former.

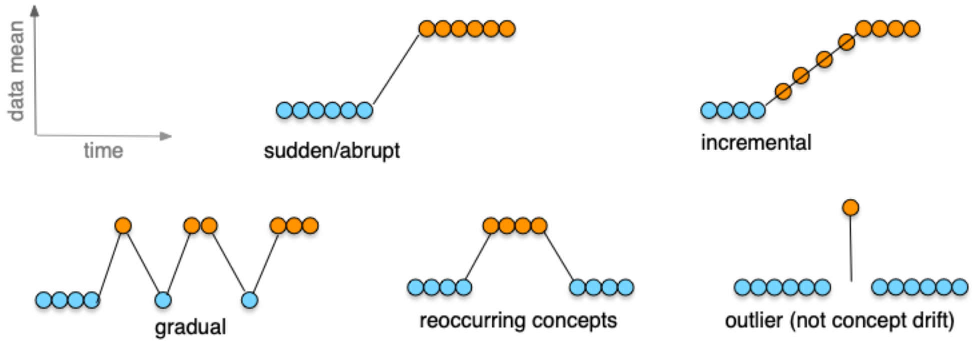

Figure 2. Patterns of changes over time (outlier is not concept drift).

Detecting drifts in practice is harder than it appears as changes can be a result of hidden variables or features that are not directly measurable [30]. A drift detection algorithm can be evaluated on the following bases [31]:

- *Differentiating* between isolated anomalies, where the underlying concept undergoes no change, and actual drift classes.
- Detecting the actual *regions* of drifts in data.
- Measuring the *severity* of drifts. The magnitude of a drift has no bearing on its classification of drift type, but for corrective purposes it carries importance.
- Robustness towards *noisy* and *imbalanced datasets*.
- Detecting *when* they occur, and *which* drift class they fall under. Notably, some drift detection algorithms can detect certain classes of drifts, but not all.

The *severity* (Δ) of a drift can be measured as $\Delta \stackrel{def}{=} \delta(P_t(X,Y), P_{t+1}(X,Y))$ [31], where δ is a function that measures the discrepancy between two data distributions, with the drift occurring between timestamps t and $t+1$.

Now, we can discuss the 'adaptation' methods that automatically adapt to drift with machine learning models, using two paradigms: *blind* and *informed*. According to Gama [29], drift adaptation by models can either be *blind* or *informed*. *Blind adaptation* refers to implicit adaptation without explicitly detecting drifts, for example, by learning on a sliding window. *Informed adaptation* refers to drifts being flagged by a detection method. Responding to informed adaptation can be done via *global replacement*, where the model is completely retrained and redeployed using the latest data or *local replacement* where the model can be adapted or partially retrained, for example, by replacing a segment of an ensemble that emits inaccurate predictions for the current concept.

2.1. Blind Adaptation

2.1.1. Sliding Window

Bachinger [27] uses a sliding window with an *offspring selection genetic algorithm* (OSGA) and a synthetic dataset for predictive manufacturing with two drift factors c and h, where c is a measurable drift and h is hidden. This approach switches the sliding window based on a selection pressure of $\frac{|Generated\ Offspring|}{|Population|}$, which outperformed polynomial regression local informed adaptation due to hidden factors, despite not having any prior knowledge regarding the measured drift factor c nor the hidden factor h.

An issue for sliding window approaches is picking the correct window size. A short window deals with immediate context but does not capture long-term dependencies. A long window captures long-term dependencies but may have different contexts embedded. Adaptive variable-size windows, which may grow or shrink, and combinations of windows have therefore gained traction [32]. An expanding or landmark window, by contrast, retains a fixed starting point and only grows.

2.1.2. Online Adaptive Learning

Online adaptive learning is a popular method using local replacement [29], that is, calculating the loss for each new prediction generated from a continuously arriving data stream, and then updating the model. More generally, *online learning* (also known as streaming learning) is a central mechanism in which new data is received for both informed and blind approaches. Jayaratne et al. [33] propose an unsupervised online learning model that can distinguish between abrupt and reoccuring drifts.

2.1.3. Ensemble Approaches

Mera et al. [34] use incremental multiple instance learning for visual inspection with an *ensemble-based* approach inspired by *Learn++*, that can learn reoccuring drifts. Ensemble learning is particularly well-suited towards reoccuring drifts since old classifiers can be reintroduced/reactivated and swapped out. For each batch of data added, a new learner is added to the ensemble, with weighted majority voting used where weights are allocated by the classifiers' performance on recent data.

2.2. Informed Adaptation

2.2.1. Error-Rate Monitoring Methods

These methods are the classical approaches in informed adaptation for supervised learning tasks. However, error rates may not be consistent if the data streams involve a lot of variability and can often miss gradual changes. *EDDM* [35] allows the identification of gradual drifts through distributions of errors, but still requires 30 classification errors, during which many samples may arrive.

2.2.2. Distribution-Based Methods

These methods involve measuring statistical properties of data and then flagging when the distribution begins to differ significantly. Bayesian autoencoders [36] focused on epistemic uncertainty which was found to be less sensitive towards sensor perturbations (noise) than the standard reconstruction loss in regular autoencoders. Distribution methods can identify the time and regions of drifts in data, but often incur a high computational cost.

2.2.3. Multiple Hypothesis Testing

This method involves identifying changes in the relationships between features and targets and forming multiple hypotheses, either hierarchically or in parallel. One example is the *linear four-rate* method [37], which monitors the true-positive rate, true-negative rate, positive predictive value, and negative predictive value, all derived from the confusion matrix. Lin et al. [38] modified this approach for imbalanced data using ensemble learning.

2.2.4. Other and Hybrid Approaches

Zenisek et al. [39] present drift detection and prediction in a time-critical environment for predictive maintenance, through a state detection model and a time-series forecasting model. Seiffer et al. [40] use SHAP-based supervised clustering for improving prediction quality of errors. Kermenov et al. [41] use a sliding window probabilistic encoder for drifting multivariate anomaly detection with industrial collaborative robots. When a drift is detected, both the detector and model return to the training stage, which comes with the drawback of increased computational cost.

Transfer Learning is often used as a secondary step of informed adaptation to map between concepts while retaining prior data. Chien et al. [42] use CNN transfer learning in the context of fault detection and classification when excursions occur (as identified by domain experts, such as ramping up with new products or novel technologies). Domain adaptation is also covered in [43,44]. McKay et al. [45] present an online transfer learning framework that is applicable for concept drifting data streams, where both the source and target domains may be online (bidirectional).

2.2.5. Label-Scarce and Unsupervised Drift Monitoring

Error-rate monitoring generally assumes that ground-truth labels become available sufficiently quickly to identify predictive degradation. This assumption may not hold in manufacturing, where labels can be scarce, expensive, or delayed until downstream inspection, maintenance, or failure. In such settings, drift may initially be monitored from the unlabelled input stream. An important distinction is that many feature-based unsupervised monitoring methods identify changes in $P(X)$, or in a representation derived from X, rather than directly establishing a change in the predictive relationship $P(Y \mid X)$. Hence, an alarm would be interpreted as evidence of a change in the monitored input distribution rather than sufficient evidence of the predictive model's invalidity.

One approach is to compare a reference window representing previously accepted operation with a recent window of incoming observations. Dos Reis et al. [46] propose an incremental Kolmogorov–Smirnov test for unsupervised online drift detection, while Gözüaçık et al. [47] formulate drift detection as a discriminative problem in which a classifier attempts to distinguish historical from recent observations. For higher-dimensional industrial data, learned representations or model uncertainty can instead be monitored. Yong et al. [36] use Bayesian autoencoders on an industrial hydraulic condition-monitoring dataset and show that epistemic uncertainty is less sensitive to sensor perturbations than conventional reconstruction loss, providing a potential signal for distinguishing changes in the operating environment from sensor-related perturbations.

Where limited labels can nevertheless be obtained, unsupervised monitoring may be combined with selective inspection or active learning. Instead of automatically adapting a model whenever a change in $P(X)$ is detected, observations from a newly detected regime can be prioritised for labelling to determine whether predictive performance has genuinely deteriorated. More generally, *active learning* approaches specifically designed for drifting data streams have been proposed to reduce labelling requirements while allowing models to respond to evolving concepts [48]. Such a strategy is particularly relevant in manufacturing, since detectable changes may arise from planned operating-mode changes, material substitutions, maintenance, or sensor recalibration without necessarily requiring model replacement.

2.3. Continual Learning

Continual learning is a sequential-learning paradigm that aims to acquire new knowledge while limiting catastrophic forgetting. It is closely related to *online learning*, although the two concepts emphasise different aspects of adaptation. Online learning concerns the incremental updating of a model as observations arrive, whereas continual learning additionally considers how knowledge acquired from previous tasks or operating conditions can be retained and exploited when learning from new ones. The two paradigms may therefore overlap, as in *online continual learning*. Likewise, continual learning does not necessarily imply explicit drift detection: it may provide the mechanism by which a model adapts after a drift has been detected, or it may update continually without identifying discrete drift points. It can therefore operate within either informed or blind adaptation.

A principal difficulty in continual learning is *catastrophic forgetting*, whereby adapting a model to newly observed data substantially degrades its performance on previously learned tasks. This is particularly relevant to smart manufacturing because changes in operating conditions are not necessarily permanent. For example, product variants, tooling configurations, materials, environmental conditions, and production schedules may recur after an intervening period. Consequently, simply overwriting knowledge associated with an earlier concept may be undesirable when that concept is subsequently encountered again. Continual learning therefore involves a *stability–plasticity trade-off*: the model must remain sufficiently plastic to learn a new operating regime while remaining sufficiently stable to preserve knowledge that continues to be useful [49].

Common continual-learning strategies include replay-based methods [50], which retain or generate representative observations from earlier regimes; regularisation-based methods, which restrict changes to parameters considered important for prior knowledge [51]; and parameter-isolation [52] or architecture-expansion methods [53], which allocate different components to different tasks or concepts. These approaches involve different trade-offs in memory, computation, model capacity, and the extent to which previous data or task identities must be retained.

Manufacturing applications have recently begun to explore this paradigm. Hua et al. [9] propose a continual-learning approach for cutting-tool wear prediction under varying cutting conditions. Their method uses a meta-LSTM that can be fine-tuned using a small number of samples for new cutting conditions and continuously updated using an orthogonal weight modification method. Maschler et al. [54] evaluate regularisation-based continual learning for anomaly detection, enabling models to adapt sequentially to changes in manufactured products while mitigating catastrophic forgetting. Also within anomaly detection, Li et al. [55] use pseudo-replay to preserve old-class knowledge while incorporating new classes.

As aforementioned, continual-learning mechanisms may be combined with either blind adaptation or explicit drift detection. Continual learning is especially suitable where concepts are *reoccuring*. More generally, it is useful where sequential adaptation is required while competence on earlier tasks or conditions should be preserved. However, determining whether a newly observed regime represents a genuinely novel concept or the return of a previous one remains non-trivial, particularly where labels are scarce or delayed. Furthermore, continual learning does not itself establish that drift has occurred. It should therefore be considered alongside the drift-monitoring approaches discussed above rather than as a replacement for them.

2.4. Discussion

Without addressing concept drift, machine learning models are susceptible to severe degradation in performance as products and their environments evolve. Measuring model performance and any degradations over time is the minimal first step towards tackling concept drift. Ideally, there is a pipeline for automatic drift detection and migration, where the migration approach taken can be determined autonomously based on the severity of the drift and the type of drift, since certain migration approaches are only suitable for certain drifts. Explicit change detection can retain more previous data and adapt more quickly in some settings [56], but this does not establish that informed adaptation dominates blind adaptation generally.

As Table 2 summarises, the choice depends principally on label latency, recurrence, and computational cost. Error-rate monitors directly reveal predictive degradation when labels arrive promptly; where labels are delayed, distribution monitors can provide earlier warnings, but a feature-distribution change need not reduce predictive performance and may miss changes in $P(Y \mid X)$. Sliding windows and online updates avoid detector design but require a defensible forgetting rate. Ensembles and transfer methods are attractive when concepts or products reoccur, provided their memory cost and the risk of negative transfer are evaluated. These families are therefore complementary rather than uniformly competing solutions.

Table 2. Comparison of principal concept-drift monitoring and adaptation families discussed in this review. The appropriate method depends on factors including label availability, recurrence, computational constraints, and whether explicit change detection is required. Notably, no family dominates in every manufacturing setting.

Method Family	**Trigger or Requirement**	**Main Strength**	**Main Limitation**	**Use Cases**
Sliding windows [27,32]	Recency rule or adaptive window; no explicit drift alarm is required	Simple blind adaptation with bounded memory; can respond rapidly to local changes	Window size determines the trade-off between responsiveness and retention; discarded observations may contain useful recurring concepts	Streams for which recent observations are most representative and model updating is inexpensive
Incremental online learning [29]	Sequential observations and an incrementally updateable model; supervised methods may require prompt labels	Continuous adaptation without complete retraining	Rapid updates may overwrite useful earlier knowledge, while errors or noise can accumulate	High-rate processes where continuous updating is appropriate and explicit drift alarms are not required
Continual learning [9,49,54,55]	A sequential stream of changing tasks, classes, domains, or operating regimes; updates may be continuous or initiated after a detected change	Explicitly addresses retention of previous knowledge while adapting to new conditions; particularly useful when earlier regimes or classes may recur	Catastrophic forgetting and the stability–plasticity trade-off remain fundamental challenges; replay, regularisation, or architectural mechanisms may add memory and computational costs, and continual learning does not itself detect drift	Changing cutting conditions, product variants, sequential anomaly-detection tasks, and new defect classes for which previously acquired knowledge remains useful
Adaptive ensembles [34]	Performance estimates for adding, weighting, retiring, or reactivating learners	Can preserve diverse or recurring concepts and replace only weak components	Additional memory and inference cost; weighting depends on representative recent evidence	Repeated products, operating modes, or seasonal concepts that may recur
Error-rate monitoring [35]	Timely ground-truth labels and a sufficiently stable error statistic	Directly detects degradation in predictive performance	Delayed or scarce labels postpone detection; noisy error sequences can obscure gradual drift	Inspection or quality-control tasks where outcomes become available promptly
Unsupervised distribution and representation monitoring [36,46,47]	Reference and recent unlabelled observations or representations, together with a discrepancy statistic or detection threshold	Can operate before ground-truth labels arrive and provide early warnings under label scarcity	A detected change in $P(X)$ or its representation does not establish a change in $P(Y \mid X)$, while real concept drift may occur without an observable marginal feature shift; high-dimensional monitoring may also be costly	Sensor and other industrial streams with delayed or scarce labels, where alarms can trigger targeted inspection or subsequent validation
Multiple-hypothesis monitoring [37,38]	Tests over several performance rates or feature–target relationships	Can identify which relationship or performance rate has changed	Multiple testing increases calibration and sample-size demands and may add computational cost	Processes where locating the affected relationship or error mode matters for intervention
Transfer and hybrid adaptation [42,45]	A related source concept or domain, often combined with a detector or expert trigger	Reuses prior knowledge and may reduce the amount of data required after a change	Source–target mismatch can cause negative transfer and must be validated	Product variants, ramp-up, or known operating transitions with related historical data

Another important element that is under-addressed in the current literature is the generation of explanations for drift. Such explanations can be verified by domain experts and can support the evaluation of automatic drift detectors by helping identify erroneous detections and confirm known drift events. Since multiple drifts may occur simultaneously with different magnitudes among multiple systems, easily distinguishing between them is of importance, especially for domain experts with no machine learning expertise. Recent work focuses on distinguishing detection from the separate tasks of locating and explaining drift [57]. For smart manufacturing in particular, there is a need for widespread adoption of approaches that mitigate forgetting, since concepts may be reoccuring as opposed to permanent and long-term changes. In this regard, *continual learning* [9,49,58] is a paradigm that warrants further investigation and deployment more broadly within the context of smart manufacturing.

3. Heterogeneous Data Streams

In smart manufacturing, there are many data streams with heterogeneity, using differing and often incompatible communication protocols and formats as well as different velocities and veracities. Moreover, the very layout of facilities and production lines can also be identical but not equivalent through upgraded equipment or other equipment differences. This contrasts with many applications of machine learning that operate under a static feature space.

Heterogeneity involves the intermingling of multiple data types and modalities (audio, image, numeric readings), statistical imbalance between resources, formats and storage structures, and levels of completeness (for example, devices that go offline due to network connections that prove unstable, causing only access to partial data), as well as dealing with devices' limited computational power resulting in larger latencies and more infrequent reporting when taking into account inference time.

Heterogeneity in data streams can be decomposed into the following categories [59,60]:

- *Semantical/conceptual*, also sometimes referred to as logical mismatch, can be divided into the following [61]:
  - *Coverage difference*: Where multiple streams model the same entity, but from different regions/areas, as is the case with a sensor measuring multiple distinct regions of a machine.
  - *Granularity difference*: Where the level of detail or fidelity differs between two data sources capturing the same entity. One sensor, for instance, may report at a more frequent interval than an identical sensor or report with higher precision. In some cases, multiresolution representations are embraced and embedded into frameworks [62,63].
  - *Perspective difference*: Where multiple data sources capture the same entity at the same granularity but from different perspectives, for example measuring humidity instead of temperature.
- *Statistical*—Different devices, production lines, or organisations may have imbalanced or non-identically distributed data at the same point in time. For instance, rare faults may be absent from some local datasets. This cross-source heterogeneity is distinct from concept, covariate, or label drift, which describes a distribution changing over time, although both can coexist.
- *Syntactical/structural*—In the case of having multiple types of sensors, whilst recording the same information, the formats, measuring units used, and communication/output data types may differ between sensors. This can be resolved with an integration layer intended to homogenise and unify all streams, the simplest being point-to-point interoperability (translating between individual files [64]). For example, manufacturing product lifecycle data (requirements, design, and quality) has domain interoperability using graphs [65].
- *Semiotic/pragmatic*—Different interpretations of an entity may exist for individuals. This type of heterogeneity is considered difficult to detect and correct.

- *Terminological* refers to the exact same features in a data source being named differently. Terminological differences can often be addressed merely by renaming and maintaining consistent naming [59], or by using a standardised common vocabulary through an ontology.

Heterogeneity as a problem extends well beyond data streams to variability in the specific tasks. Some approaches to that end include *multi-task learning*, *active learning* and more generally *meta-learning*, and *curriculum learning*.

Since smart manufacturing incorporates varying layouts, variability in speed and execution is expected, where heterogeneity is introduced from underlying equipment itself and sensors with different time resolutions. For time-series data in particular, techniques such as *Dynamic Time Warping* [8,66] may be used to align data, whereby the similarity between sequences of varying lengths and speeds may be measured and used in applications such as clustering and classification.

Multiple types of heterogeneity often occur simultaneously. For manufacturing, semantical and statistical heterogeneity are particularly significant: the latter arises when devices, lines, or sites have different data distributions, independently of any temporal drift. In the following sections, we will discuss three approaches to dealing with and integrating heterogeneous data streams: federated learning, multimodal sensor fusion, and heterogeneous transfer learning.

3.1. Semantic Interoperability

In this section, we discuss semantic interoperability through standardised information models of how industrial assets and their data are represented. *OPC UA* supports domain-specific information models through Companion Specifications. For example, OPC UA for Machine Tools defines a common interface across machine tools of different technologies, manufacturers, and model series, exposing information for monitoring and production management [67]. This allows analytics applications to consume a common representation rather than resorting entirely to vendor-specific mappings.

The *Asset Administration Shell* (AAS) provides a complementary asset-oriented representation organised into semantically defined submodels [68]. Similarly, the AAS Predictive Maintenance submodel standardises the representation of information such as predicted remaining useful life and failure probability without prescribing the underlying prediction model [69]. Hence different machines may use different prognostic models while exposing their outputs through a common representation.

3.2. Federated Learning

Federated learning (FL) is a distributed learning technique, which differs from the typically used *centralised learning* (CL). There are three main types of federated learning, in terms of their relationship to feature spaces and sample identities [70]:

- *Horizontal federated learning* uses similar feature spaces across parties with different sample or entity populations.
- *Vertical federated learning* applies when parties have overlapping sample identities but hold different feature sets. Raw samples are not thereby shared, and which party holds labels depends on the protocol.
- *Federated transfer learning* applies when both feature spaces and sample populations differ, enabling cross-domain applications [71,72].

In cloud manufacturing, however, data protection is often required between production lines [73] when outsourcing is involved. A key advantage of FL is that raw records need not be centralised, which can reduce exposure. FL does not itself provide a formal privacy guarantee because shared gradients or model updates can leak training information [74];

additional controls such as secure aggregation, differential privacy, or cryptographic computation must be selected for an explicit threat model. Subject to such controls, when production is outsourced to another facility that has capacity/a spare production line, optimisation can still occur without trade secret data being directly shared. The benefits of privacy-enhanced federated learning among small and medium-sized enterprises are discussed in [75] to resolve data quality issues of imbalanced datasets and low data quantities that, without collaboration between enterprises, prohibit any form of reliable decision-making model from being developed by any single organisation. *Model poisoning*, where a malicious actor sends manipulated updates to compromise the global model, is also a security concern for organisations and SMEs that wish to use external updates, where there is ongoing research for robust FL in the IIoT [76].

Often data is also locally stored and captured in a distributed manner, so centralising data can cause latency due to the necessity of data transmission. Specialised heterogeneous or personalised FL methods can accommodate differences in client distributions and, in some cases, model architectures [77]. Model federation alone, however, does not reconcile units, schemas, feature meanings, communication protocols, or terminology; these require preprocessing, explicit alignment, ontologies, or suitable heterogeneous-FL protocols.

Many classical distributed-optimisation analyses assume independent and identically distributed data. FL clients, however, commonly have different contemporaneous distributions. This cross-client statistical heterogeneity is distinct from temporal distribution drift.

Statistical heterogeneity has been tackled before, in terms of dealing with *non-IID* data [78]. One approach is *personalised learning* by modelling the problem as *multi-task learning*. For example, a robust global model can be trained with the non-IID and unbalanced distributions in different participants [77]. There is also some need to automatically detect and quantify statistical heterogeneity, such as through techniques like local dissimilarity [79]. Ang et al. [80] discuss robustness towards noise in federated learning.

Coverage difference refers to when an individual sensor may measure only a partial state of the system and therefore be insufficient for system-level inference [24]. Separately, heterogeneous local objectives can complicate global FL optimisation; gradient-negotiation and neighbour-consensus approaches for industrial systems are discussed in [81].

3.3. Multimodal Sensor Fusion

Data *modality* refers to the handling of different data types, such as *unstructured data* like images and video (which may be used for quality control prediction and adjustment via imaging) and audio (which is often used to detect subtle faults). *Structured data* includes standardised numeric sensor readings that capture machine operational parameters and environmental data. *Semi-structured data* also exists and has organisational properties, such as tags and metadata, but does not stick to a fixed schema, such as log files and inspection data for machines that may carry information regarding operational status, performance, and errors. Modalities are categories defined by how a particular data type is received, represented, and interpreted [82].

In this context, sensor fusion refers to combining data from different sources and improving the knowledge extracted from them. Early fusion combines raw data or extracted features, whereas late fusion combines the results of different algorithms at the final stage of analysis [82]. Where there are multiple modalities in conjunction with sensor fusion, it is then referred to as *multimodal sensor fusion* or *multisensory integration*.

The importance of multimodal integration is reflected in the development of ML architectures designed explicitly to process heterogeneous modalities, such as the *Perceiver*, which uses Transformer-style attention while accommodating high-dimensional inputs from different modalities [83]. More recently, *foundation models*—models pretrained on large corpora of diverse data at scale and subsequently adaptable to a range of downstream tasks—have extended beyond language to multimodal and time-series applications [84]. Time-series foundation models are particularly relevant to manufacturing: *MOMENT* supports general-purpose tasks including forecasting, classification, anomaly detection, and imputation [85], while *TimesFM* demonstrates zero-shot transfer for forecasting across previously unseen time-series datasets [86]. Such approaches may be useful where manufacturing environments contain many related sensor streams but comparatively little labelled data for individual machines or operating conditions. However, the adoption of foundation models in manufacturing remains comparatively limited. Industrial data are often specialised and heterogeneous across machines, processes, and modalities, while industrial applications impose stringent requirements for trustworthy outputs and domain-specific adaptation [87]. In addition, large and standardised industrial pretraining datasets remain relatively scarce due to data being commercially sensitive, which limits the development of manufacturing-specific foundation models.

Generally, there are three levels of fusion:

- *Data-based*: If the data has a similar distribution and format/type, a combined matrix can be formed. However, considering that data may come at different intervals such as sensor readings and an image at the end-stage for quality classification, this approach does not really capture the heterogeneity required in smart manufacturing.
- *Feature-based*: High-level features are extracted and then combined. For example, a CNN or vision transformer may handle images, a recurrent neural network may be used for time-series sensor readings, and then, a fusion layer is introduced with the high-level representations. The fusion may be: *early-stage*, *late-stage*, or even *intermediate*.
- *Decision-based*: Decisions are made as a result of multiple individual models. The limitation is that only local state is captured, and there is no cross-modal awareness. The weights applied to each model can then be calculated by cross-validation, particle swarm optimisation, or genetic algorithms.

Fu et al. [26] use a multimodal neural network for multisensory fault diagnosis, based on 'dynamic routing', with multimodal feature extraction to generate a representation from the different sources hierarchically. Kounta et al. [88] use late-stage fusion for predicting the optimal cutting parameters in milling with digital input data processed by an LSTM network and images processed by a CNN network.

Rahate et al. [25] discuss sensor fusion in the presence of missing modalities, that is, when one of the modalities is partially or completely missing, as is the case with real-world equipment. Noisy modalities can also be addressed through techniques such as data augmentation, in which noise is introduced during training to improve robustness to noisy conditions; however, this involves a trade-off between accuracy and robustness.

When considering semantic perspective heterogeneity, multimodality may be useful. For example, optimising a production process could have multiple factors: an imaging quality control system as well as a manual rating. Faults may be detected using audio data in unison with sensor readings. Instead of considering single sources of truth independently, multiple are present.

Where local devices perform early fusion, this can eliminate the need for redundant communication to a centralised server, which takes time because complete raw data is sent prior to feature extraction.

3.4. Heterogeneous Transfer Learning

Heterogeneous transfer learning is a form of transfer learning [89] involving a source task s and a target task t, where the feature spaces may differ ($\mathcal{X}_s \neq \mathcal{X}_t$) and the output spaces may also differ ($\mathcal{Y}_s \neq \mathcal{Y}_t$). The source and target tasks may additionally have different marginal or conditional probability distributions, such that $(P(x_s) \neq P(x_t)) \vee (P(y_s \mid x_s) \neq P(y_t \mid x_t))$. It can be leveraged for addressing semantical drifts, for example, modelling differences between two plants that produce the same product with a slightly differing facility layout: sensors mounted in different places due to physical constraints, variants or substitutes of equipment (even in terms of configurations), and so on.

Statistical heterogeneity can also be alleviated by heterogeneous transfer learning (with a focus on differing probability distributions), since the source task may be focused on optimising for one product variant or set of factors whereas the target task is focused on a different variant, varying in materials and dimensions, due to having different specifications in one plant. The latter case may have a shortage of data; therefore, augmenting data and learning general properties through heterogeneous transfer learning may immensely improve performance.

Negative transfer is a possibility when domains differ too much such that the source domain has little relevance, or may even be more harmful than using traditional training methods. A further issue is *transfer stability*. As discussed in Section 2, it is not uncommon that source and target domains undergo changes themselves throughout training; therefore substantial modifications may be required since the assumption of static domains may not hold. Essentially, where domains have considerable discrepancies, the generation of a domain-invariant space is non-trivial. As such, a major issue with heterogeneous transfer learning is that there is no guarantee that it will be successful with complex tasks and complex variations in domains [90], which is an intrinsic issue to heterogeneous data streams.

On the other hand, there are a few measures and techniques [91] that can be used that do assist with making heterogeneous transfer learning more successful instead of having negative transfer, for example, instance weighting/resampling, applying more importance to instances in $\mathcal{D}_s$ that resemble $\mathcal{D}_t$, which may in particular be useful where marginal probability distributions differ between the source and target. Alternatively, shared samples can be used, termed as co-occurrence data [92].

Domain adaptation is a related subtype of transfer learning [89], where the source and target typically share a feature space but differ in their distributions; heterogeneous transfer learning does not impose that shared-feature-space requirement.

3.5. Discussion

Without addressing heterogeneous data streams, the models that are generated will not be robust against any changes in production layouts (for example, changing production lines that have dissimilar elements) or facility changes. Table 3 discusses the approaches that we mentioned against the types of heterogeneity that we introduced.

Some types of heterogeneity may be addressed relatively easily through foresight by the system integrator. Where heterogeneity is primarily syntactical, terminological, or semantic, it is preferable to resolve it at the integration layer where possible, for example through common information models, ontologies, OPC UA Companion Specifications, or AAS submodels. A machine learning model should not be required to infer that two differently named or structured variables represent the same physical quantity when this correspondence can be explicitly represented.

The three learning paradigms address different layers of the problem rather than serving as substitutes. Federated learning is appropriate when data remain distributed across organisations or sites, but it does not by itself align schemas or meanings. Multimodal sensor fusion combines complementary observations of the same process, but requires temporal, spatial, and semantic alignment. Heterogeneous transfer learning is useful when knowledge must cross feature spaces or domains, but its benefit depends on source–target relatedness and must be checked against negative transfer. A deployment may therefore combine an integration layer with fusion, transfer, or federation according to the heterogeneity present.

Table 3. Heterogeneitydimensions directly addressed by the reviewed approaches. A checkmark indicates that an approach directly addresses at least one substantive aspect of the corresponding dimension; a cross indicates that it is not a primary mechanism for addressing that dimension.

	Semantical	Statistical	Syntactical	Semiotic	Terminological
Semantic interoperability	✓	×	✓	×	✓
Federated learning	×	✓	×	×	×
Multimodal sensor fusion	✓	×	×	×	×
Heterogeneous transfer learning	✓	✓	×	×	×

4. Trustworthy Systems

This paper considers three capabilities of machine learning: namely, *prediction* (associative learning ($P(Y \mid X)$)), *control* (taking actions to achieve desired outcomes), and *explanation* (which involves counterfactuals/hypotheticals). These are not an exhaustive taxonomy of machine learning tasks. Trustworthy systems go beyond standard predictive models, but their validity, reliability, safety, robustness, transparency, privacy, and fairness must be evaluated for the intended context [93].

Trustworthy models do not inherently provide robustness and correctness in the dynamic setting of manufacturing; these properties must be evaluated. Where a decision-making support system is *'black-box'*, human operators may rely on their own intuition and ignore it because they cannot understand or verify its outputs.

With respect to this, we essentially have to consider the following aspects: integration of domain knowledge, explainability, and modelling of uncertainty.

4.1. Integration of Domain Knowledge

Domain knowledge has bidirectional implications between raw data and a priori knowledge, in the sense that augmenting a data-driven model with domain knowledge may increase the performance or robustness of the model, but actionable insights may also be gained through domain knowledge that is learnt (or 'extracted') through data.

Such knowledge is accumulated by experts with decades of experience. Datasets are known to be expensive to generate, and often do not capture knowledge to its fullest extent: for example, there may be edge cases and rare faults or issues addressed before they had a chance to develop and present in data (i.e., preemptive actions instead of focusing on learning corrective or remedial behaviour) that an ML system may not be aware of. Optimal parameters and generalisation can be greatly enhanced through incorporating domain knowledge instead of ignoring it.

Integration has many forms: Ref. [94] embeds discovered prior knowledge into a hybrid feedforward neural network as nonlinear constraints or a penalty function for learning the base neural network, increasing process safety and generalisability. Marazopoulou et al. [95] integrate prior domain knowledge insofar as partial ordering of variables. Fu et al. [26] use a Hilbert transform to add a priori knowledge. *Simulation-*

informed learning uses outputs from physics-based simulations as additional training data or features [96,97].

A particularly interesting emerging area is *neurosymbolic AI*, which combines 'deductive' (symbolic) and 'inductive' (neural) models. For example, neural networks paired with a semantic network/knowledge-base have been combined to control the quality of product labelling [98]. Saleeshya and Binu [99] use a neuro-fuzzy hybrid model for the assessment of leanness of manufacturing systems. Van [100] use non-experts to repair reinforcement learning policies for robots after failures and generate shields for future corrections.

4.1.1. Language Models

Language models, combined with techniques such as *retrieval augmented generation* [101], are particularly relevant since factories possess extensive natural-language knowledge in manuals, work instructions, and issue reports. Notably, early manufacturing studies integrating language models [102,103] find that language model assistants are useful as a supplement to human expertise. Users generally preferred asking a nearby experienced colleague when one was available [104]. In particular, language models currently lack correctness guarantees, which is not favourable when incorrect or inadequate answers could create a safety risk. Moreover, the knowledge base incurs a maintenance cost to keep it current, sufficiently detailed and factory-specific [103].

4.1.2. Knowledge Discovery

Data can be used in order to extract useful knowledge that has wider implications in aspects such as facility and equipment choices, analysing causes of observations. Kamm et al. [59] provide a review for knowledge discovery for heterogeneous and unstructured data.

4.1.3. Rule Learning

Rule mining refers to learning association rules from data; this can involve the consideration of sometimes hundreds of rules, with varying degrees of usefulness. With that said, rule mining is a way of gaining insights from data. Chen et al. [105] use association rule mining for defect detection. Djatna [106] develops a maintenance strategy from 83 qualified rules that were mined to find the relationship between overall equipment effectiveness and the response of action required given the current condition. Fuzzy rules, to deal with impreciseness, can also be mined, as presented in [107,108].

4.1.4. Pattern Recognition

Pattern recognition refers to extracting knowledge from data, with the intent of identifying patterns and useful information. For example, Soualhi et al. [109] identify faults by splitting the data into clusters in a fuzzy manner. Clusters are classified and compared with a reference pattern, namely healthy operation of the system, to identify classes. By looking at parameters set by process operators, data mining has even been used to develop a meta-controller in aluminium processing that integrates all process parameters [110,111].

Process mining models process flows through event log data [112,113]. Petri nets have been derived from event traces and logs [114], which can support reliability assessment [115].

Statistical pattern analysis identifies deviations and similarities in system behaviour based on the statistical characteristics of observed data [116]. For example, this approach has been applied to fault detection in semiconductor batch processes [117].

4.1.5. Composite Event Recognition (CER)

This is the detection of composite activities or events based on simple events derived from time-series data, where there may be both temporal and atemporal constraints. Mantenoglou et al. [118] present online event recognition over noisy data streams. Kapp et al. [119] present pattern recognition in multivariate time-series. There has also been work on integrating source identification in event-based recognition for transient tailbacks [120].

4.1.6. Stream Mining

Stream mining refers to a variant of data mining that extracts patterns from data that is continuously and rapidly provided in real-time, considering the dynamic nature of such a stream in relation to issues such as concept drift. Torkamani and Lohweg [121] provide a survey on *motif discovery* within time-series data, where *motifs* refer to recurring patterns and sequences. This may then be used to find patterns that indicate machine failure or patterns that correspond to variations in product quality.

4.1.7. Structural Learning

Structural learning refers to learning some structure, purely from data. While it is a computationally intensive task, structural learning has seen applications. Learning the dependency graph of a Bayesian belief network solely from data is NP-hard; methods for doing so are surveyed in [122]. Salmani and Katoen [123] proposes automatically computing inference probabilities for Bayesian networks through probabilistic model checking.

Wang [24] presents an approach that mines a dependency graph for anomalies, which is a directed graph $\langle V, E\rangle$ where each element of V represents an anomalous event and elements of E show the temporal dependencies between two vertices. This approach naturally lends itself well to explainability. Fault trees have been automatically extracted from time-series data [124].

4.1.8. Causal Discovery

Causal discovery refers to discovering underlying causal relationships among variables in a particular system or dataset. In particular, causal analysis is necessitated not only for robustness or trustworthiness, but dually for optimisations: inquisitive and speculative queries (instead of predictive) such as "if I use material X instead of material Y" or perhaps facility layout X/Y, does product quality improve or worsen on the whole?

Zhou et al. [125] uses a causal knowledge graph for root cause analysis of equipment spot inspection failures. Yang et al. [126] use causal graphical modelling for detecting blockages, including disturbances that are not observable, for a steel casting process.

Marazopoulou et al. [95] use Causal Bayesian Networks for causal discovery in manufacturing to improve product yield. For time-series analysis in particular, there is also the notion of *temporal precedence* where causes precede their effects [127].

4.2. Explainability

An *explainable AI* (XAI) decision-making support system allows highlighting the relevant information in order to make the best informed decisions. Such explanations can also enhance the verifiability of AI-advised decisions (as long as the resulting explanations are sufficiently informative [128]), which is particularly important because contemporary models may generalise poorly and underperform on out-of-distribution samples.

XAI has seen application within smart manufacturing, although its adoption remains limited in practice [129]. In processes such as injection molding, defects are rare, leading to highly imbalanced datasets that make it difficult to fine-tune process parameters. *Shapley*

Additive Explanations (SHAP) [130] has been used in this context to identify the process variables most influential in the model's defect predictions [131]. SHAP assigns each input feature a contribution score to a given prediction, based on concepts from cooperative game theory. In automated visual inspection of industrial products, Zhou et al. [132] use a Double-VGG16 CNN to classify defect types and then apply *Grad-CAM* [133] to localise the defects, checking where the network "looked" when labelling a scratch or pit.

Interpretability depends on the model structure, representation, audience, and purpose [134]. Small decision trees, logistic regression, or linear SVMs may support intrinsic interpretation in suitable settings, whereas nonlinear kernel SVMs, complex Bayesian models, and neural networks can be opaque and may require *post hoc* explanations. Interpretable models do not universally sacrifice predictive accuracy and can be competitive in some applications [135]. Post hoc methods can be *model-agnostic*, such as SHAP, LIME [136], and partial dependence plots [137], or *model-specific*, such as feature importance for tree ensembles. *Counterfactual explanations* [138] are an explanation type for which particular generation algorithms may be model-agnostic or model-specific.

The scope of post hoc explanations is also a consideration. *Local explanations* justify a particular prediction or a particular subset of inputs, whereas *global explanations* explain the model's behaviour as a whole and generally, for example through summarising counterfactual rules [139] within deep neural networks.

More broadly within machine learning, having high-quality explanations is an active research area. A few criteria have been established towards evaluating explanation quality formally [140–142]:

- *Fidelity* captures the property of faithfulness. An explanation has high fidelity if, given only the explanation, the model's behaviour can be approximated in the relevant region of the input space. Practically, if a root cause analysis depicts 'spindle vibration' and 'feed rate' as factors that drive scrap, but the model is driven by an unobserved proxy (e.g., a timestep), interventions will fail.
- *Sparsity* is about how few elements (features, rules, time steps, spatial regions) the explanation uses while still remaining faithful. Sparse explanations are easier for engineers to read and act on, but overly sparse explanations can misrepresent the model.
- *Stability* measures how sensitive explanations are to small changes in input, data, or model initialisation. If two very similar parts or process states get very different explanations, engineers will doubt the system, even if predictions are good. Another important element of stability is temporal stability; given some drift (see Section 2), explanations should also evolve smoothly. Note that stability carries information not captured by fidelity alone [143].

Some approaches add model interpretability or model visualisation, which are useful in designing model architectures or tuning hyperparameters, but not for providing high-level explanations to end-users. The intersection of domain knowledge (see Section 4.1) with explainability is an active research area, for instance, validating ML decisions against causal knowledge graphs [144].

4.3. Representing Uncertainty

Two widely used categories are *aleatoric uncertainty*, arising from irreducible variability or observation noise, and *epistemic uncertainty*, arising from limited knowledge of the model or its parameters and potentially reducible with informative data [145]. Distribution drift is not itself epistemic uncertainty, but it can increase epistemic uncertainty when a model trained on an earlier distribution no longer represents the current process. Likewise, an *out-of-distribution* (OOD) sample lies outside the training distribution and may increase model uncertainty, but OOD observations can occur without temporal drift, and drift does not make every new observation OOD.

For models with a specified predictive-variance decomposition, the law of total variance can separate predictive variance into aleatoric and epistemic components. This additive identity is not universal across all uncertainty measures or model classes. In high-stakes processes, such as aerospace manufacturing [146] or chemical production [147], calibrated probabilities or prediction intervals can help trigger human review or fail-safes. In the following sections, we cover a few approaches towards modelling uncertainty.

4.3.1. Probabilistic

Probabilistic approaches can represent observation variability and, when uncertainty over model parameters is included, epistemic uncertainty. Bayesian inference combines a prior and likelihood to form a posterior; in sequential inference, one posterior may serve as the prior for the next update. *Variational inference* is often used to approximate otherwise intractable posterior distributions.

4.3.2. Bayesian Belief Networks

These probabilistic graphical models represent relationships among random variables $\mathbf{V}$. A Bayesian network comprises a directed acyclic graph $G = \langle \mathbf{V}, \mathbf{E} \rangle$ and a conditional distribution $P(V_i \mid \mathrm{Pa}(V_i))$ for each node given all of its parents, yielding the factorisation $P(\mathbf{V}) = \prod_i P(V_i \mid \mathrm{Pa}(V_i))$. A causal interpretation requires additional assumptions. Jones et al. [148] use Bayesian-network modelling for maintenance planning by analysing failure rates and probabilities. *Dynamic Bayesian networks*, which represent discrete timesteps, have been used in manufacturing for reliability analysis [149] and estimating process availability [150].

4.3.3. Bayesian Deep Learning

Bayesian neural networks place prior distributions over weights or other model parameters and infer posterior distributions from data, allowing predictive uncertainty to be estimated by marginalising over parameter uncertainty [151]. Bayesian autoencoders for drift detection based on epistemic uncertainty are presented in [36].

4.3.4. Fuzzy and Rough Approaches

Fuzzy approaches refer to techniques that use fuzzy logic or fuzzy set theory to deal with imprecision, uncertainty, and ambiguity. Fuzzy logic extends classical logic by replacing discrete truth values of $\{0, 1\}$ with degrees of truth in the interval $[0, 1]$, where variables are in fuzzy sets based on degrees of membership instead of binary (also referred to as 'crisp') membership. A review of fuzzy logic in manufacturing is provided by Azadegan et al. [152]. Hong et al. [153] use fuzzy c-means (where data points are assigned to a cluster, with a certain degree) for concept-drift patterns.

Unlike fuzzy sets, *rough sets* require that objects are either members or non-members of a set, but with a boundary region additionally based on a lower and upper approximation of potential members of sets, which have seen use in quality control [154].

Fuzzy–rough approaches connect the two [155]. For example, [156] uses a rough–fuzzy approach for supply chain selection, where fuzzy sets handle internal uncertainty and rough sets handle external uncertainty.

Fuzzy approaches deal with degrees of truth (imprecision and vagueness), whereas probabilistic methods focus on the likelihood of events (randomness and variability). In this regard, approaches that combine fuzzy and probabilistic methods have also been proposed. For example, Ocampo [157] uses a probabilistic fuzzy analytic network process for sustainable manufacturing strategy decisions. Gul et al. [158] combine fuzzy and probabilistic risk analysis for manufacturing failure mode and effect analysis, while Djelloul et al. [159] combine a neuro-fuzzy approach with a probabilistic model for fault diagnosis.

4.3.5. Uncertainty Sampling

Uncertainty sampling can find cases where a machine learning model is uncertain in its prediction—indicating an area of focus for data scientists and domain experts by highlighting a potential weakness in the model, whether because of a lack of data or perhaps design limitations (e.g., lack of sensors providing necessary data). This technique quantifies uncertainty but does not by itself validate the reliability of the model. It is often used within *active learning* to focus on the most important data samples. Adaptive weighted uncertainty sampling for active learning in additive manufacturing is presented in [160].

4.3.6. Conformal Prediction

Conformal prediction wraps a predictive model to produce prediction sets or intervals. Given a miscoverage level ϵ, standard conformal methods provide finite-sample *marginal* coverage of at least $1 - \epsilon$ under exchangeability and the conditions of the chosen conformal procedure [161]. This guarantee concerns repeated exchangeable observations; it is not generally a conditional probability guarantee for one particular part or input. Conformal methods are otherwise agnostic to the underlying predictive model. Exchangeability means that the joint distribution is invariant to reordering of the observations. Revealing previous labels before subsequent predictions does not establish exchangeability. Temporal dependence or distribution drift can violate it, in which case weighted, adaptive, or other non-exchangeable conformal methods and their corresponding guarantees are required [162].

Akpabio [163] uses conformal prediction for uncertainty quantification in line edge roughness. Javanmardi and Hüllermeier [164] use conformal prediction for remaining useful life estimation. Boursinos and Koutsoukos [165] perform assurance monitoring of learning-enabled cyber-physical systems using inductive conformal prediction.

4.4. Discussion

The need for trustworthy systems is substantial, due to the fact that a key requirement of manufacturing is reliability. Without reliable solutions, the industry crumbles under increased labour costs and inability to meet the needs of clients. One first step would be to aim to represent uncertainty, which would be useful in recognising system shortcomings as well as potentially dealing with issues presented in Section 2, namely concept drift. Then, if possible, explainability solutions should be evaluated. Encoding domain knowledge is another step towards providing a fallback against reliability issues, in particular with out-of-distribution data samples.

Table 4 makes clear that these capabilities are complementary. An intrinsically interpretable model is preferable when it meets the task's performance requirements; post hoc explanations require separate fidelity and stability checks when an opaque model is justified. Uncertainty estimates indicate the strength or limits of a prediction but do not explain its cause, while explanations do not provide calibrated risk. Domain knowledge can constrain implausible behaviour, but it must itself be validated and maintained. Consequently, no individual method establishes trustworthiness: the appropriate combination depends on the decision risk, available feedback, and assumptions that can be monitored after deployment.

Table 4. Comparison of method families that support trustworthy use. Notice that the method families provide *complementary* advantages.

Method Family	Primary Role	Requirement or Assumption	Main Limitation	Use Cases
Domain-knowledge integration [94,96]	Constrain learning or supplement sparse data with rules, causal structure, or simulations	Encoded knowledge or simulations must be relevant and sufficiently accurate	Incorrect, incomplete, or obsolete knowledge can bias the model; knowledge maintenance and conflict resolution remain necessary	Safety constraints or rare cases are known but poorly represented in data
Intrinsically interpretable models [134,135]	Make model reasoning inspectable through a suitable model structure or representation	The task must admit an interpretable model with adequate predictive performance for the intended audience	Interpretability does not establish correctness or causality, and complex data may resist a simple faithful representation	A transparent model can meet the operational performance requirement
Post hoc explanations [130,136]	Explain predictions from an already-trained, potentially opaque model	The explanation method, background data, and scope must match the model and decision context	Fidelity, stability, and usefulness require separate evaluation; feature attribution does not establish physical causation	An opaque model is operationally justified and individual decisions require review
Probabilistic and Bayesian methods [145,151]	Represent predictive distributions and, for specified models, parameter uncertainty	A defensible probabilistic model, inference procedure, and calibration assessment	Misspecification, prior sensitivity, approximate inference, and computational cost can produce misleading uncertainty	Risk decisions require probabilities or intervals and model assumptions can be checked
Fuzzy and rough approaches [152,154]	Represent vagueness, imprecision, or boundary regions	Membership functions, rules, or approximation relations must be elicited or learned	Results depend on representation choices and do not by themselves quantify event frequency or predictive calibration	Expert concepts have graded or indeterminate boundaries
Uncertainty sampling [160]	Prioritise uncertain cases for labelling or expert inspection	The uncertainty score must be meaningful and labels or expert feedback must be obtainable	Selecting uncertain samples does not validate the model or guarantee reliable uncertainty estimates	Labelling capacity is limited and can be directed to informative cases
Conformal prediction [161,162]	Wrap a predictive model with prediction sets or intervals having a coverage guarantee	Standard guarantees require exchangeability and the conditions of the chosen conformal procedure	Standard marginal coverage is not pointwise conditional coverage; temporal dependence and drift require adapted methods and guarantees	Operational decisions benefit from coverage-controlled sets or intervals and assumptions can be monitored

5. Findings, Conclusions and Outlook

In this paper, we have outlined some of the most pressing issues limiting widespread adoption of smart manufacturing, and some of the emerging and actively ongoing research into solutions for these real-world challenges. We discussed all of these great challenges both at the conceptual level and in the currently proposed solutions. This is a narrative review and does not claim systematic or exhaustive coverage.

To address the issue of dynamic manufacturing processes that undergo continuous changes, we introduce the problem as concept drift and the two main categories of solutions: *blind* and *informed* adaptation. We also find that *continual learning* [49] is a growing paradigm but has seen limited applications within the manufacturing industry so far, and exhibits promising potential. Heterogeneous data streams, due to the various sensor devices and manufacturing equipment, which are always being upgraded and replaced, add a great deal of complexity. However, solutions are available, including heterogeneous federated learning, multimodal sensor fusion and heterogeneous transfer learning. Nevertheless, these techniques are advanced and necessitate additional modelling effort in smart manufacturing.

Then, we discussed trustworthy manufacturing systems, to allow a symbiotic relationship between production operatives and the smart manufacturing models. We recommend integrating representation of model uncertainty: this is substantially simpler than fully-fledged explanations and provides an accessible step towards explainable AI.

Integration of domain knowledge is also particularly important in the context of smart manufacturing, where neuro-symbolic AI solutions offer promise.

By considering the issues (and their respective solutions) we have posed in this paper early in the design process, data scientists and AI engineers will be equipped to deal with conceptual and large-scale issues that are uniformly present in the domain of manufacturing. As we have discussed in this survey, manufacturing involves a dynamic and evolving environment that must adapt flexibly to market demand and supply chain issues, with real-world equipment and sensors that are fallible and noisy. As such, a continual learning paradigm, in the online or streaming setting, may mitigate performance degradation due to operational changes and disturbances, but does not guarantee non-diminishing accuracy because catastrophic forgetting, the stability–plasticity trade-off, resource constraints, and changing task definitions require empirical evaluation [49].

Active areas of research include the integration of domain knowledge in conjunction with explainability. The combination of these two elements may improve some dimensions of model trustworthiness. Moreover, the consideration of explainability and domain knowledge must be done in relation to the other challenges of concept drift and heterogeneous data streams, rather than independently. While other survey papers cover smart manufacturing in a more general sense, and overview the existing state-of-the-art of technologies that bring manufacturing to the digital era, this paper specifically focuses on the grand challenges that arise at deployment time and beyond, and discusses the long-term usage of smart manufacturing solutions with regard to aspects such as how to maintain optimal performance even when the underlying process evolves (e.g., ingredient and parameter changes), and how production operatives can effectively use, diagnose, and understand artificial intelligence solutions.

Author Contributions: Conceptualization, O.A. and S.K.; writing—original draft, O.A.; writing—review & editing, N.P. and S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by funding from Innovate UK grant no: 10028947 (Made Smarter Innovation: Sustainable Smart Factory).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analysed in this study. Data sharing is not applicable to this article.

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

References

1. Hoffmann, J.B.; Heimes, P.; Senel, S. IoT platforms for the Internet of production. *IEEE Internet Things J.* **2018**, *6*, 4098–4105. [CrossRef]
2. Cook, A.A.; Misirli, G.; Fan, Z. Anomaly Detection for IoT Time-Series Data: A Survey. *IEEE Internet Things J.* **2020**, *7*, 6481–6494. [CrossRef]
3. Nagorny, K.; Lima-Monteiro, P.; Barata, J.; Colombo, A.W. Big data analysis in smart manufacturing: A review. *Int. J. Commun. Netw. Syst. Sci.* **2017**, *10*, 31–58.
4. Çınar, Z.M.; Abdussalam Nuhu, A.; Zeeshan, Q.; Korhan, O.; Asmael, M.; Safaei, B. Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0. *Sustainability* **2020**, *12*, 8211. [CrossRef]
5. Pearl, J. Theoretical impediments to machine learning with seven sparks from the causal revolution. *arXiv* **2018**, arXiv:1801.04016.
6. Wu, D.; Greer, M.J.; Rosen, D.W.; Schaefer, D. Cloud manufacturing: Strategic vision and state-of-the-art. *J. Manuf. Syst.* **2013**, *32*, 564–579. [CrossRef]
7. Ren, L.; Zhang, L.; Wang, L.; Tao, F.; Chai, X. Cloud manufacturing: Key characteristics and applications. *Int. J. Comput. Integr. Manuf.* **2017**, *30*, 501–515. [CrossRef]

8. Kim, H.; Ahn, C.R.; Engelhaupt, D.; Lee, S. Application of dynamic time warping to the recognition of mixed equipment activities in cycle time measurement. *Autom. Constr.* **2018**, *87*, 225–234. [CrossRef]
9. Hua, J.; Li, Y.; Mou, W.; Liu, C. An accurate cutting tool wear prediction method under different cutting conditions based on continual learning. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2022**, *236*, 123–131. [CrossRef]
10. Ziekow, H.; Schreier, U.; Gerling, A.; Saleh, A. Interpretable Machine Learning for Quality Engineering in Manufacturing-Importance Measures that Reveal Insights on Errors. In Proceedings of the Upper-Rhine Artificial Intelligence Symposium, UR-AI 2021, Artificial Intelligence-Application in Life Sciences and Beyond, Kaiserslautern, Germany, 27 October 2021; pp. 96–105.
11. Li, C.; Zheng, P.; Yin, Y.; Wang, B.; Wang, L. Deep reinforcement learning in smart manufacturing: A review and prospects. *CIRP J. Manuf. Sci. Technol.* **2023**, *40*, 75–101. [CrossRef]
12. Schwung, D.; Reimann, J.N.; Schwung, A.; Ding, S.X. Smart manufacturing systems: A game theory based approach. In *Intelligent Systems: Theory, Research and Innovation in Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 51–69.
13. Kusiak, A. Smart manufacturing. *Int. J. Prod. Res.* **2018**, *56*, 508–517. [CrossRef]
14. Kusiak, A. Fundamentals of smart manufacturing: A multi-thread perspective. *Annu. Rev. Control* **2019**, *47*, 214–220. [CrossRef]
15. Zheng, P.; Wang, H.; Sang, Z.; Zhong, R.Y.; Liu, Y.; Liu, C.; Mubarok, K.; Yu, S.; Xu, X. Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives. *Front. Mech. Eng.* **2018**, *13*, 137–150. [CrossRef]
16. Li, D.; Liu, S.; Wang, B.; Yu, C.; Zheng, P.; Li, W. Trustworthy AI for human-centric smart manufacturing: A survey. *J. Manuf. Syst.* **2025**, *78*, 308–327. [CrossRef]
17. Deokar, S.; Kumar, N.; Singh, R.P. A comprehensive review on smart manufacturing using machine learning applicable to fused deposition modeling. *Results Eng.* **2025**, *26*, 104941. [CrossRef]
18. Benhanifia, A.; Cheikh, Z.B.; Oliveira, P.M.; Valente, A.; Lima, J. Systematic review of predictive maintenance practices in the manufacturing sector. *Intell. Syst. Appl.* **2025**, *26*, 200501. [CrossRef]
19. Bandhana, A.; Vokřínek, J. AI-Driven Manufacturing: Surveying for Industry 4.0 and Beyond. *Oper. Res. Forum* **2025**, *6*, 145. [CrossRef]
20. Ramesh, K.; Indrajith, M.N.; Prasanna, Y.S.; Deshmukh, S.S.; Parimi, C.; Ray, T. Comparison and assessment of machine learning approaches in manufacturing applications. *Ind. Artif. Intell.* **2025**, *3*, 2. [CrossRef]
21. Chhetri, T.R.; Aghaei, S.; Fensel, A.; Göhner, U.; Gül-Ficici, S.; Martinez-Gil, J. Optimising Manufacturing Process with Bayesian Structure Learning and Knowledge Graphs. In Proceedings of the Computer Aided Systems Theory–EUROCAST 2022: 18th International Conference, Las Palmas de Gran Canaria, Spain, 20–25 February 2022; pp. 594–602.
22. Hauder, V.A.; Beham, A.; Wagner, S.; Doerner, K.F.; Affenzeller, M. Dynamic online optimization in the context of smart manufacturing: An overview. *Procedia Comput. Sci.* **2021**, *180*, 988–995. [CrossRef]
23. Blömeke, S.; Rickert, J.; Mennenga, M.; Thiede, S.; Spengler, T.S.; Herrmann, C. Recycling 4.0–Mapping smart manufacturing solutions to remanufacturing and recycling operations. *Procedia CIRP* **2020**, *90*, 600–605. [CrossRef]
24. Wang, J.; Liu, C.; Zhu, M.; Guo, P.; Hu, Y. Sensor Data Based System-Level Anomaly Prediction for Smart Manufacturing. In Proceedings of the 2018 IEEE International Congress on Big Data (BigData Congress), San Francisco, CA, USA, 2–7 July 2018; pp. 158–165. [CrossRef]
25. Rahate, A.; Mandaokar, S.; Chandel, P.; Walambe, R.; Ramanna, S.; Kotecha, K. Employing multimodal co-learning to evaluate the robustness of sensor fusion for industry 5.0 tasks. *Soft Comput.* **2023**, *27*, 4139–4155. [CrossRef]
26. Fu, P.; Wang, J.; Zhang, X.; Zhang, L.; Gao, R.X. Dynamic routing-based multimodal neural network for multi-sensory fault diagnosis of induction motor. *J. Manuf. Syst.* **2020**, *55*, 264–272. [CrossRef]
27. Bachinger, F.; Kronberger, G.; Affenzeller, M. Continuous improvement and adaptation of predictive models in smart manufacturing and model management. *IET Collab. Intell. Manuf.* **2021**, *3*, 48–63. [CrossRef]
28. Lyu, M.; Li, X.; Chen, C.H. Achieving Knowledge-as-a-Service in IIoT-driven smart manufacturing: A crowdsourcing-based continuous enrichment method for Industrial Knowledge Graph. *Adv. Eng. Inform.* **2022**, *51*, 101494. [CrossRef]
29. Gama, J.; Žliobaitė, I.; Bifet, A.; Pechenizkiy, M.; Bouchachia, A. A survey on concept drift adaptation. *ACM Comput. Surv.* **2014**, *46*, 44:1–44:37. [CrossRef] [PubMed]
30. Liu, A.; Song, Y.; Zhang, G.; Lu, J. Regional Concept Drift Detection and Density Synchronized Drift Adaptation. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 2280–2286. [CrossRef] [PubMed]
31. Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; Zhang, G. Learning under Concept Drift: A Review. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 2346–2363. [CrossRef]
32. Yun, U.; Lee, G. Sliding window based weighted erasable stream pattern mining for stream data applications. *Future Gener. Comput. Syst.* **2016**, *59*, 1–20. [CrossRef]
33. Jayaratne, D.; De Silva, D.; Alahakoon, D.; Yu, X. Continuous detection of concept drift in industrial cyber-physical systems using closed loop incremental machine learning. *Discov. Artif. Intell.* **2021**, *1*, 7. [CrossRef]

34. Mera, C.; Orozco-Alzate, M.; Branch, J. Incremental learning of concept drift in Multiple Instance Learning for industrial visual inspection. *Comput. Ind.* **2019**, *109*, 153–164. [CrossRef]
35. Baena-Garcıa, M.; del Campo-Ávila, J.; Fidalgo, R.; Bifet, A.; Gavalda, R.; Morales-Bueno, R. Early drift detection method. In *Proceedings of the Fourth International Workshop on Knowledge Discovery from Data Streams*; Association for Computing Machinery: New York, NY, USA, 2006; Volume 6, pp. 77–86.
36. Yong, B.X.; Fathy, Y.; Brintrup, A. Bayesian autoencoders for drift detection in industrial environments. In *Proceedings of the 2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT*; IEEE: New York, NY, USA, 2020; pp. 627–631.
37. Wang, H.; Abraham, Z. Concept drift detection for streaming data. In *Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN)*; IEEE: New York, NY, USA, 2015; pp. 1–9.
38. Lin, C.C.; Deng, D.J.; Kuo, C.H.; Chen, L. Concept drift detection and adaption in big imbalance industrial IoT data using an ensemble learning method of offline classifiers. *IEEE Access* **2019**, *7*, 56198–56207. [CrossRef]
39. Zenisek, J.; Holzinger, F.; Affenzeller, M. Machine learning based concept drift detection for predictive maintenance. *Comput. Ind. Eng.* **2019**, *137*, 106031. [CrossRef]
40. Seiffer, C.; Ziekow, H.; Schreier, U.; Gerling, A. Detection of Concept Drift in Manufacturing Data with SHAP Values to Improve Error Prediction. *Data Anal.* **2021**, 51–60.
41. Kermenov, R.; Nabissi, G.; Longhi, S.; Bonci, A. Anomaly Detection and Concept Drift Adaptation for Dynamic Systems: A General Method with Practical Implementation Using an Industrial Collaborative Robot. *Sensors* **2023**, *23*, 3260. [CrossRef] [PubMed]
42. Chien, C.F.; Hung, W.T.; Liao, E.T.Y. Redefining monitoring rules for intelligent fault detection and classification via CNN transfer learning for smart manufacturing. *IEEE Trans. Semicond. Manuf.* **2022**, *35*, 158–165. [CrossRef]
43. Halstead, B.; Koh, Y.S.; Riddle, P.; Pears, R.; Pechenizkiy, M.; Bifet, A.; Olivares, G.; Coulson, G. Analyzing and repairing concept drift adaptation in data stream classification. *Mach. Learn.* **2022**, *111*, 3489–3523. [CrossRef]
44. Karimian, M.; Beigy, H. Concept drift handling: A domain adaptation perspective. *Expert Syst. Appl.* **2023**, *224*, 119946. [CrossRef]
45. McKay, H.; Griffiths, N.; Taylor, P.; Damoulas, T.; Xu, Z. Online transfer learning for concept drifting data streams. In *Proceedings of the 8th International Workshop on Big Data, IoT Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications Conference*; ACM: New York, NY, USA, 2019; Volume 2579.
46. Dos Reis, D.M.; Flach, P.; Matwin, S.; Batista, G. Fast unsupervised online drift detection using incremental kolmogorov-smirnov test. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1545–1554.
47. Gözüaçık, Ö.; Büyükçakır, A.; Bonab, H.; Can, F. Unsupervised concept drift detection with a discriminative classifier. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2365–2368.
48. Žliobaitė, I.; Bifet, A.; Pfahringer, B.; Holmes, G. Active learning with drifting streaming data. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *25*, 27–39. [CrossRef] [PubMed]
49. Wang, L.; Zhang, X.; Su, H.; Zhu, J. A comprehensive survey of continual learning: Theory, method and application. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 5362–5383. [CrossRef] [PubMed]
50. Lopez-Paz, D.; Ranzato, M. Gradient episodic memory for continual learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA*; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 6470–6479.
51. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 3521–3526. [CrossRef] [PubMed]
52. Mallya, A.; Lazebnik, S. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2018; pp. 7765–7773.
53. Rusu, A.A.; Rabinowitz, N.C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; Hadsell, R. Progressive neural networks. *arXiv* **2016**, arXiv:1606.04671.
54. Maschler, B.; Pham, T.T.H.; Weyrich, M. Regularization-based continual learning for anomaly detection in discrete manufacturing. *Procedia CIRP* **2021**, *104*, 452–457. [CrossRef]
55. Li, Y.; Xie, T.; Liu, C.; Shi, Z. Pseudo replay-based class continual learning for online new category anomaly detection in advanced manufacturing. *IISE Trans.* **2025**, *57*, 1407–1421. [CrossRef]
56. Klinkenberg, R.; Renz, I. Adaptive Information Filtering: Learning in the Presence of Concept Drifts. In *AAAI-98 Workshop on Learning for Text Categorization*; 1998; pp. 33–40. Available online: https://cdn.aaai.org/Workshops/1998/WS-98-05/WS98-05-006.pdf (accessed on 15 July 2026).
57. Hinder, F.; Vaquet, V.; Hammer, B. One or two things we know about concept drift—A survey on monitoring in evolving environments. Part A: Detecting concept drift. *Front. Artif. Intell.* **2024**, *7*, 1330257. [CrossRef] [PubMed]

58. Liu, B. Learning on the job: Online lifelong and continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Palo Alto, CA, USA, 2020; Volume 34, pp. 13544–13549.
59. Kamm, S.; Jazdi, N.; Weyrich, M. Knowledge Discovery in Heterogeneous and Unstructured Data of Industry 4.0 Systems: Challenges and Approaches. *Procedia CIRP* **2021**, *104*, 975–980. [CrossRef]
60. Jirkovskỳ, V.; Obitko, M.; Mařík, V. Understanding data heterogeneity in the context of cyber-physical systems integration. *IEEE Trans. Ind. Inform.* **2016**, *13*, 660–667. [CrossRef]
61. Jirkovskỳ, V.; Obitko, M. Semantic Heterogeneity Reduction for Big Data in Industrial Automation. In *ITAT 2014 with Selected Papers from Znalosti 2014, CEUR Workshop Proceedings Vol. 1214*; 2014; Volume 1214. Available online: https://ceur-ws.org/Vol-1214/z1.pdf (accessed on 15 July 2026).
62. Ulieru, M.; Norrie, D.; Kremer, R.; Shen, W. A multi-resolution collaborative architecture for web-centric global manufacturing. *Inf. Sci.* **2000**, *127*, 3–21. [CrossRef]
63. Kang, S.; Jeon, J.; Kim, H.S.; Chun, I. CPS-based fault-tolerance method for smart factories. *Automatisierungstechnik* **2016**, *64*, 750–757. [CrossRef]
64. Hedberg, T., Jr.; Feeney, A.B.; Helu, M.; Camelio, J.A. Toward a lifecycle information framework and technology in manufacturing. *J. Comput. Inf. Sci. Eng.* **2017**, *17*, 021010. [CrossRef] [PubMed]
65. Hedberg, T.D., Jr.; Bajaj, M.; Camelio, J.A. Using graphs to link data across the product lifecycle for enabling smart manufacturing digital threads. *J. Comput. Inf. Sci. Eng.* **2020**, *20*, 011011. [CrossRef]
66. Ratanamahatana, C.A.; Keogh, E. Everything you know about dynamic time warping is wrong. In *Proceedings of the Third Workshop on Mining Temporal and Sequential Data*; Citeseer: University Park, PA, USA, 2004; Volume 32.
67. Mahnke, W.; Leitner, S.H.; Damm, M. *OPC Unified Architecture*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 1.
68. Ye, X.; Hong, S.H. Toward industry 4.0 components: Insights into and implementation of asset administration shells. *IEEE Ind. Electron. Mag.* **2019**, *13*, 13–25. [CrossRef]
69. Cavalieri, S.; Salafia, M.G. A model for predictive maintenance based on Asset Administration Shell. *Sensors* **2020**, *20*, 6028. [CrossRef] [PubMed]
70. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* **2019**, *10*, 1–19. [CrossRef]
71. Kevin, I.; Wang, K.; Zhou, X.; Liang, W.; Yan, Z.; She, J. Federated transfer learning based cross-domain prediction for smart manufacturing. *IEEE Trans. Ind. Inform.* **2021**, *18*, 4088–4096. [CrossRef]
72. Feng, S.; Li, B.; Yu, H.; Liu, Y.; Yang, Q. Semi-Supervised Federated Heterogeneous Transfer Learning. *Knowl.-Based Syst.* **2022**, *252*, 109384. [CrossRef]
73. Ge, N.; Li, G.; Zhang, L.; Liu, Y. Failure prediction in production line based on federated learning: An empirical study. *J. Intell. Manuf.* **2022**, *33*, 2277–2294. [CrossRef]
74. Zhu, L.; Liu, Z.; Han, S. Deep Leakage from Gradients. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2019; Volume 32, pp. 14747–14756.
75. Zhang, J.; Cooper, C.; Gao, R.X. Federated Learning for Privacy-Preserving Collaboration in Smart Manufacturing. In *Manufacturing Driving Circular Economy: Proceedings of the 18th Global Conference on Sustainable Manufacturing, October 5–7, 2022, Berlin*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 845–853.
76. Zhang, J.; Ge, C.; Hu, F.; Chen, B. Robustfl: Robust federated learning against poisoning attacks in industrial iot systems. *IEEE Trans. Ind. Inform.* **2021**, *18*, 6388–6397. [CrossRef]
77. Gao, D.; Yao, X.; Yang, Q. A Survey on Heterogeneous Federated Learning. *arXiv* **2022**, arXiv:2210.04505.
78. Huang, W.; Ye, M.; Du, B. Learn from Others and Be Yourself in Heterogeneous Federated Learning. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 10133–10143. [CrossRef]
79. Wu, H.; Wang, P. Fast-convergent federated learning with adaptive weighting. *IEEE Trans. Cogn. Commun. Netw.* **2021**, *7*, 1078–1088. [CrossRef]
80. Ang, F.; Chen, L.; Zhao, N.; Chen, Y.; Wang, W.; Yu, F.R. Robust federated learning with noisy communication. *IEEE Trans. Commun.* **2020**, *68*, 3452–3464. [CrossRef]
81. Savazzi, S.; Nicoli, M.; Bennis, M.; Kianoush, S.; Barbieri, L. Opportunities of Federated Learning in Connected, Cooperative and Automated Industrial Systems. *IEEE Commun. Mag.* **2021**, *59*, 16–21. [CrossRef]
82. Tsanousa, A.; Bektsis, E.; Kyriakopoulos, C.; González, A.G.; Leturiondo, U.; Gialampoukidis, I.; Karakostas, A.; Vrochidis, S.; Kompatsiaris, I. A Review of Multisensor Data Fusion Solutions in Smart Manufacturing: Systems and Trends. *Sensors* **2022**, *22*, 1734. [CrossRef] [PubMed]
83. Jaegle, A.; Gimeno, F.; Brock, A.; Vinyals, O.; Zisserman, A.; Carreira, J. Perceiver: General perception with iterative attention. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2021; pp. 4651–4664.

84. Bommasani, R.; Hudson, D.A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M.S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. On the opportunities and risks of foundation models. *arXiv* **2021**, arXiv:2108.07258.
85. Goswami, M.; Szafer, K.; Choudhry, A.; Cai, Y.; Li, S.; Dubrawski, A. Moment: A family of open time-series foundation models. *arXiv* **2024**, arXiv:2402.03885.
86. Das, A.; Kong, W.; Sen, R.; Zhou, Y. A decoder-only foundation model for time-series forecasting. *arXiv* **2023**, arXiv:2310.10688.
87. Ren, L.; Wang, H.; Dong, J.; Jia, Z.; Li, S.; Wang, Y.; Laili, Y.; Huang, D.; Zhang, L.; Li, B. Industrial foundation model. *IEEE Trans. Cybern.* **2025**, *55*, 2286–2301. [CrossRef] [PubMed]
88. Kounta, C.A.K.A.; Kamsu-Foguem, B.; Noureddine, F.; Tangara, F. Multimodal deep learning for predicting the choice of cut parameters in the milling process. *Intell. Syst. Appl.* **2022**, *16*, 200112. [CrossRef]
89. Day, O.; Khoshgoftaar, T.M. A survey on heterogeneous transfer learning. *J. Big Data* **2017**, *4*, 29. [CrossRef]
90. Yan, R.; Shen, F.; Sun, C.; Chen, X. Knowledge transfer for rotary machine fault diagnosis. *IEEE Sens. J.* **2019**, *20*, 8374–8393. [CrossRef]
91. Niu, S.; Liu, Y.; Wang, J.; Song, H. A decade survey of transfer learning (2010–2020). *IEEE Trans. Artif. Intell.* **2020**, *1*, 151–166. [CrossRef]
92. Yang, L.; Jing, L.; Yu, J.; Ng, M.K. Learning transferred weights from co-occurrence data for heterogeneous transfer learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *27*, 2187–2200. [CrossRef] [PubMed]
93. Tabassi, E. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*; Technical Report NIST AI 100-1; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023. [CrossRef]
94. Haichuan, L.; Hongye, S.; Lei, X.; Yong, G.; Gang, R. Multiple-prior-knowledge neural network for industrial processes. In *Proceedings of the 2010 IEEE International Conference on Automation and Logistics*; IEEE: New York, NY, USA, 2010; pp. 385–390.
95. Marazopoulou, K.; Ghosh, R.; Lade, P.; Jensen, D. Causal Discovery for Manufacturing Domains. *arXiv* **2016**, arXiv:1605.04056. [CrossRef]
96. Wang, J.; Li, Y.; Gao, R.X.; Zhang, F. Hybrid physics-based and data-driven models for smart manufacturing: Modelling, simulation, and explainability. *J. Manuf. Syst.* **2022**, *63*, 381–391. [CrossRef]
97. Chen, J.; Pierce, J.; Williams, G.; Simpson, T.W.; Meisel, N.; Prabha Narra, S.; McComb, C. Accelerating thermal simulations in additive manufacturing by training physics-informed neural networks with randomly synthesized data. *J. Comput. Inf. Sci. Eng.* **2024**, *24*, 011004. [CrossRef]
98. Golovko, V.; Kroshchanka, A.; Kovalev, M.; Taberko, V.; Ivaniuk, D. Neuro-symbolic artificial intelligence: Application for control the quality of product labeling. In *Proceedings of the International Conference on Open Semantic Technologies for Intelligent Systems*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 81–101.
99. Saleeshya, P.G.; Binu, M. A neuro-fuzzy hybrid model for assessing leanness of manufacturing systems. *Int. J. Lean Six Sigma* **2019**, *10*, 473–499. [CrossRef]
100. van Waveren, S.; Pek, C.; Tumova, J.; Leite, I. Correct me if I'm wrong: Using non-experts to repair reinforcement learning policies. In *Proceedings of the 2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*; IEEE: New York, NY, USA, 2022; pp. 493–501.
101. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, M.; Wang, H. Retrieval-augmented generation for large language models: A survey. *arXiv* **2023**, arXiv:2312.10997.
102. Bajestani, M.S.; Mun, D.; Kim, D.B. Human-in-the-loop and large language models in smart manufacturing: Current applications, challenges, and perspectives. *J. Manuf. Syst.* **2026**, *86*, 913–941. [CrossRef]
103. Maghanaki, M.; Shahin, M.; Chen, F.F. Large language models in manufacturing: A comprehensive review. *Int. J. Adv. Manuf. Technol.* **2026**, 1–27. [CrossRef]
104. Freire, S.K.; Wang, C.; Foosherian, M.; Wellsandt, S.; Ruiz-Arenas, S.; Niforatos, E. Knowledge sharing in manufacturing using large language models: User evaluation and model benchmarking. *arXiv* **2024**, arXiv:2401.05200.
105. 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]
106. 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]
107. Altuntas, S.; Dereli, T.; Selim, H. Fuzzy weighted association rule based solution approaches to facility layout problem in cellular manufacturing system. *Int. J. Ind. Syst. Eng.* **2013**, *15*, 253–271. [CrossRef]
108. Vinodh, S.; Prakash, N.H.; Selvan, K.E. Evaluation of leanness using fuzzy association rules mining. *Int. J. Adv. Manuf. Technol.* **2011**, *57*, 343–352. [CrossRef]
109. Soualhi, M.; Nguyen, K.T.; Medjaher, K. Pattern recognition method of fault diagnostics based on a new health indicator for smart manufacturing. *Mech. Syst. Signal Process.* **2020**, *142*, 106680. [CrossRef]
110. Kusiak, A. A data mining approach for generation of control signatures. *J. Manuf. Sci. Eng.* **2002**, *124*, 923–926. [CrossRef]
111. Kusiak, A. Data mining: Manufacturing and service applications. *Int. J. Prod. Res.* **2006**, *44*, 4175–4191. [CrossRef]

112. Van Der Aalst, W. *Process Mining: Data Science in Action*; Springer: Berlin/Heidelberg, Germany, 2016; Volume 2.
113. Lorenz, R.; Senoner, J.; Sihn, W.; Netland, T. Using process mining to improve productivity in make-to-stock manufacturing. *Int. J. Prod. Res.* **2021**, *59*, 4869–4880. [CrossRef]
114. Leemans, S.J.; Fahland, D.; Van Der Aalst, W.M. Discovering block-structured process models from event logs-a constructive approach. In *Proceedings of the International Conference on Applications and Theory of Petri Nets and Concurrency*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 311–329.
115. Friederich, J.; Lazarova-Molnar, S. Data-Driven Reliability Modeling of Smart Manufacturing Systems Using Process Mining. In *Proceedings of the 2022 Winter Simulation Conference (WSC)*; IEEE: New York, NY, USA, 2022; pp. 2534–2545.
116. Wang, J.; He, Q.P. Multivariate statistical process monitoring based on statistics pattern analysis. *Ind. Eng. Chem. Res.* **2010**, *49*, 7858–7869. [CrossRef]
117. He, Q.P.; Wang, J. Statistics pattern analysis: A new process monitoring framework and its application to semiconductor batch processes. *AIChE J.* **2011**, *57*, 107–121. [CrossRef]
118. Mantenoglou, P.; Artikis, A.; Paliouras, G. Online Event Recognition over Noisy Data Streams. *Int. J. Approx. Reason.* **2023**, *161*, 108993. [CrossRef]
119. Kapp, V.; May, M.C.; Lanza, G.; Wuest, T. Pattern recognition in multivariate time series: Towards an automated event detection method for smart manufacturing systems. *J. Manuf. Mater. Process.* **2020**, *4*, 88. [CrossRef]
120. Schwenke, C.; Wagner, T.; Gellrich, A.; Kabitzsch, K. Event-based recognition and source identification of transient tailbacks in manufacturing plants. In *Proceedings of the 2012 Winter Simulation Conference (WSC)*; IEEE: New York, NY, USA, 2012; pp. 1–12.
121. Torkamani, S.; Lohweg, V. Survey on time series motif discovery. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2017**, *7*, e1199. [CrossRef]
122. Scanagatta, M.; Salmerón, A.; Stella, F. A survey on Bayesian network structure learning from data. *Prog. Artif. Intell.* **2019**, *8*, 425–439. [CrossRef]
123. Salmani, B.; Katoen, J.P. Automatically Finding the Right Probabilities in Bayesian Networks. *J. Artif. Intell. Res.* **2023**, *77*, 1637–1696. [CrossRef]
124. Lazarova-Molnar, S.; Niloofar, P.; Barta, G.K. Data-Driven Fault Tree Modeling for Reliability Assessment of Cyber-Physical Systems. In Proceedings of the 2020 Winter Simulation Conference (WSC), Orlando, FL, USA, 14–18 December 2020; pp. 2719–2730, ISSN 1558-4305. [CrossRef]
125. Zhou, B.; Li, J.; Li, X.; Hua, B.; Bao, J. Leveraging on causal knowledge for enhancing the root cause analysis of equipment spot inspection failures. *Adv. Eng. Inform.* **2022**, *54*, 101799. [CrossRef]
126. Yang, S.; Rebmann, A.; Tang, M.; Moravec, R.; Behrmann, D.; Baird, M.; Bequette, B.W. Process monitoring using causal graphical models, with application to clogging detection in steel continuous casting. *J. Process Control* **2021**, *105*, 259–266. [CrossRef]
127. Eichler, M. Causal inference in time series analysis. In *Causality: Statistical Perspectives and Applications*; John Wiley & Sons Inc.: Hoboken, NJ, USA, 2012; pp. 327–354.
128. Fok, R.; Weld, D.S. In Search of Verifiability: Explanations Rarely Enable Complementary Performance in AI-Advised Decision Making. *arXiv* **2023**, arXiv:2305.07722.
129. Puthanveettil Madathil, A.; Luo, X.; Liu, Q.; Walker, C.; Madarkar, R.; Qin, Y. A review of explainable artificial intelligence in smart manufacturing. *Int. J. Prod. Res.* **2025**, *63*, 8654–8697. [CrossRef]
130. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA*; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 4768–4777.
131. Hong, J.; Hong, Y.; Baek, J.W.; Kang, S.W. Enhancing the Product Quality of the Injection Process Using eXplainable Artificial Intelligence. *Processes* **2025**, *13*, 912. [CrossRef]
132. Zhou, F.; Liu, G.; Xu, F.; Deng, H. A generic automated surface defect detection based on a bilinear model. *Appl. Sci.* **2019**, *9*, 3159. [CrossRef]
133. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
134. Lipton, Z.C. The Mythos of Model Interpretability. *Commun. ACM* **2018**, *61*, 36–43. [CrossRef]
135. Rudin, C. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nat. Mach. Intell.* **2019**, *1*, 206–215. [CrossRef] [PubMed]
136. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.

137. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
138. Leofante, F.; Wicker, M. *Robust Explainable AI*; Springer: Berlin/Heidelberg, Germany, 2025.
139. Rawal, K.; Lakkaraju, H. Beyond Individualized Recourse: Interpretable and Interactive Summaries of Actionable Recourses. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 12187–12198.
140. Dai, J.; Upadhyay, S.; Aivodji, U.; Bach, S.H.; Lakkaraju, H. Fairness via explanation quality: Evaluating disparities in the quality of post hoc explanations. In Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society, Oxford, UK, 19–21 May 2022; pp. 203–214.
141. Nazir, M.A.; Evangelista, E.; Bukhari, S.M.S.; Sharma, R. A survey of feature attribution techniques in explainable AI: Taxonomy, analysis and comparison. *Ann. Math. Comput. Sci.* **2025**, *28*, 115–126. [CrossRef]
142. Mersha, M.; Lam, K.; Wood, J.; Alshami, A.K.; Kalita, J. Explainable artificial intelligence: A survey of needs, techniques, applications, and future direction. *Neurocomputing* **2024**, *599*, 128111. [CrossRef]
143. Ballegeer, M.; Bogaert, M.; Benoit, D.F. Evaluating the stability of model explanations in instance-dependent cost-sensitive credit scoring. *Eur. J. Oper. Res.* **2025**, *326*, 630–640. [CrossRef]
144. Jaimini, U.; Sheth, A. Causalkg: Causal knowledge graph explainability using interventional and counterfactual reasoning. *IEEE Internet Comput.* **2022**, *26*, 43–50. [CrossRef]
145. Kendall, A.; Gal, Y. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2017; Volume 30, pp. 5574–5584.
146. Zhang, Y. Efficient Uncertainty Quantification in Aerospace Analysis and Design. Ph.D. Thesis, Missouri University of Science and Technology, Rolla, MI, USA, 2013.
147. Vishwakarma, G.; Sonpal, A.; Hachmann, J. Metrics for benchmarking and uncertainty quantification: Quality, applicability, and best practices for machine learning in chemistry. *Trends Chem.* **2021**, *3*, 146–156. [CrossRef]
148. Jones, B.; Jenkinson, I.; Yang, Z.; Wang, J. The use of Bayesian network modelling for maintenance planning in a manufacturing industry. *Reliab. Eng. Syst. Saf.* **2010**, *95*, 267–277. [CrossRef]
149. Weber, P.; Jouffe, L. Reliability modelling with dynamic bayesian networks. *IFAC Proc. Vol.* **2003**, *36*, 57–62. [CrossRef]
150. Weber, P. Dynamic bayesian networks model to estimate process availability. In Proceedings of the 8th International Conference Quality, Reliability, Maintenance, CCF'02, Sinaia, Romania, 18–20 September 2002; MEDIAREX 21; pp. 184–189.
151. Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; Wierstra, D. Weight Uncertainty in Neural Network. In *Proceedings of the 32nd International Conference on Machine Learning*; JMLR.org: Norfolk, MA, USA, 2015; Volume 37, pp. 1613–1622.
152. Azadegan, A.; Porobic, L.; Ghazinoory, S.; Samouei, P.; Kheirkhah, A.S. Fuzzy logic in manufacturing: A review of literature and a specialized application. *Int. J. Prod. Econ.* **2011**, *132*, 258–270. [CrossRef]
153. Hong, T.P.; Chen, C.H.; Li, Y.K.; Wu, M.T. Using Fuzzy C-means to Discover Concept-drift Patterns for Membership Functions. *Trans. Fuzzy Sets Syst.* **2022**, *1*, 21–31.
154. Chien, C.F.; Wu, H.J. Integrated circuit probe card troubleshooting based on rough set theory for advanced quality control and an empirical study. *J. Intell. Manuf.* **2022**, *35*, 275–287. [CrossRef]
155. Radzikowska, A.M.; Kerre, E.E. A comparative study of fuzzy rough sets. *Fuzzy Sets Syst.* **2002**, *126*, 137–155. [CrossRef]
156. Chen, Z.; Ming, X.; Zhou, T.; Chang, Y. Sustainable supplier selection for smart supply chain considering internal and external uncertainty: An integrated rough-fuzzy approach. *Appl. Soft Comput.* **2020**, *87*, 106004. [CrossRef]
157. Ocampo, L. A probabilistic fuzzy analytic network process approach (PROFUZANP) in formulating sustainable manufacturing strategy infrastructural decisions under firm size influence. *Int. J. Manag. Sci. Eng. Manag.* **2018**, *13*, 158–174. [CrossRef]
158. Gul, M.; Yucesan, M.; Celik, E. A manufacturing failure mode and effect analysis based on fuzzy and probabilistic risk analysis. *Appl. Soft Comput.* **2020**, *96*, 106689. [CrossRef]
159. Djelloul, I.; Sari, Z.; Latreche, K. Uncertain fault diagnosis problem using neuro-fuzzy approach and probabilistic model for manufacturing systems. *Appl. Intell.* **2018**, *48*, 3143–3160. [CrossRef]
160. van Houtum, G.J.; Vlasea, M.L. Active learning via adaptive weighted uncertainty sampling applied to additive manufacturing. *Addit. Manuf.* **2021**, *48*, 102411. [CrossRef]
161. Shafer, G.; Vovk, V. A Tutorial on Conformal Prediction. *J. Mach. Learn. Res.* **2008**, *9*, 371–421.
162. Barber, R.F.; Candes, E.J.; Ramdas, A.; Tibshirani, R.J. Conformal prediction beyond exchangeability. *Ann. Stat.* **2023**, *51*, 816–845. [CrossRef]
163. Akpabio, I.I. Uncertainty Quantification in Line Edge Roughness Estimation Using Conformal Prediction. Master's Thesis, Texas A&M University, College Station, TX, USA, 2022.

164. Javanmardi, A.; Hüllermeier, E. Conformal Prediction Intervals for Remaining Useful Lifetime Estimation. *arXiv* **2022**, arXiv:2212.14612.
165. Boursinos, D.; Koutsoukos, X. Assurance monitoring of learning-enabled cyber-physical systems using inductive conformal prediction based on distance learning. *AI EDAM* **2021**, *35*, 251–264. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

https://doi.org/10.3390/app16178599